

Technical University of Denmark



## Competitive Liner Shipping Network Design

**Karsten, Christian Vad; Pisinger, David; Røpke, Stefan**

*Publication date:*  
2015

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Karsten, C. V., Pisinger, D., & Røpke, S. (2015). Competitive Liner Shipping Network Design. Kgs. Lyngby: DTU Management Engineering.

**DTU Library**  
Technical Information Center of Denmark

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Competitive Liner Shipping Network Design

Christian Vad Karsten



Kongens Lyngby 2015

Technical University of Denmark  
DTU Management Engineering  
Management Science  
Produktionstorvet, building 426,  
2800 Kongens Lyngby, Denmark  
Phone +45 4525 4800  
[www.man.dtu.dk](http://www.man.dtu.dk)

# Summary (English)

---

The goal of this thesis is to develop decision support tools, which can be used to optimize container shipping networks while supporting competitive transportation services. The competitiveness of container liner shipping is to a high degree determined by transportation times and number of transshipments on the most important sailing routes. The proposed methods in this thesis, aimed at liner shipping network design, integrate competitiveness such that the fuel consumption per transported container is reduced without increasing the transit times.

A well-designed route net is decisive for container shipping company earnings. The operation of the route net constitute the majority of the total costs, so it is essential to achieve a good capacity utilization in a route plan with travel times that satisfy customer requirements. Most academic articles dealing with the design of container networks neither take the container transportation times that can be realized in the network nor the number of transshipments into consideration. This is mainly because the optimization problem is based on other transportation networks where these constraints are not decisive to the quality of the network. Furthermore, the problem in itself is challenging to optimize due to its size and complexity. However, the field has seen crucial progress and is mature to include handling of competitiveness in the actual design of the network.

As a liner shipping network is an organic entity, which is constantly changed to reflect changes in the freight markets, it is of significant value that the changes are based on the existing network, which presumably is of high quality. At the same time, changes often affect a limited geographical area in the global shipping market. In this thesis methods to incorporate the competitiveness of the network in the form of requested transportation times and transshipments and to ensure better capacity utilization in the network are presented. The project has developed large-scale mathematical methods that leverage the existing network to optimize a specific freight market/geographic area or the

entire network. The result is prototypes of decision support tools to make incremental changes to a network e.g. by adding/deleting ports from routes or change speed between two ports in order to examine how it changes the total earnings taking into account the network's competitiveness and quality.

The contributions of this thesis cover modeling, methodology, and applications. The developed methods address operational (cargo routing), tactical (speed optimization and service selection), and strategic (network design) planning problems faced by liner shipping companies. Ultimately, the proposed methods help answer questions such as:

How can the capacity utilization of the network be improved while taking into account the competitiveness and quality?

How should new routes be designed such that they utilize existing and new markets or possibly leave unprofitable markets?

What routes are the most profitable to operate?

How should changes in the fleet be integrated into the existing network?

What ships will be relevant to use in the future?

What transportation times and number of transshipments would be appropriate to provide for a given transport?

# Resumé (Summary in Danish)

---

Målet for denne afhandling er at udvikle beslutningsstøtteværktøjer, der kan bruges til at optimere containerrederiers rutenetværk, samtidig med at de understøtter konkurrencedygtige transportydelser. Konkurrencedygtigheden af containerlinjefart afgøres i høj grad af transporttider og antal omladninger (transshipments) på de vigtigste ruter. De foreslåede metoder til design af linjefartsnetværk i denne afhandling integrerer hensynet til konkurrencedygtighed, således at brændstofforbruget per transporteret container mindskes uden at øge transporttiderne.

Et veludviklet rutenet er afgørende for indtjeningen i containerrederier. Driften af et rutenet udgør langt størstedelen af de samlede omkostninger, så det er essentielt at opnå en god kapacitetsudnyttelse i en ruteplan med transporttider, der opfylder kundernes behov. De fleste videnskabelige artikler, som beskæftiger sig med design af containernetværk, tager ikke højde for hverken de transporttider, der kan realiseres i netværket, eller antal omladninger. Dette skyldes, at optimeringsproblemet har taget sit udgangspunkt i andre transportnetværk, hvor disse begrænsninger ikke er afgørende for kvaliteten af netværket, samtidig med at problemet i sig selv har været vanskeligt at optimere grundet dets størrelse og kompleksitet. Der er imidlertid sket afgørende fremskridt indenfor feltet, som nu er modent til at indarbejde håndtering af konkurrencedygtighed i det faktiske design af rutenetværket.

Da netværket er en organisk størrelse, der løbende justeres til ændringer i fragtmarkedet, er der stor værdi i at ændringerne tager udgangspunkt i det eksisterende netværk, der formentlig allerede er af høj kvalitet. Ændringer vil ofte påvirke et afgrænset geografisk område i det globale fragtmarked. I denne afhandling præsenteres metoder til at indarbejde konkurrencedygtigheden af netværket, i form af de efterspurgte transporttider og omladninger, samt sikre

en bedre kapacitetsudnyttelse i netværket. Der er i projektet blevet udarbejdet storskala matematiske metoder, der gør det muligt, med udgangspunkt i det eksisterende netværk, at optimere netværket både for et specifikt fragtmarked/geografisk område og det samlede rutenetværk. Resultatet er prototyper af beslutningsstøtteværktøjer til at foretage inkrementelle ændringer i netværket f.eks. ved at tilføje/slette havne fra ruter eller ændre hastighed mellem to havne med henblik på at belyse, hvorledes det ændrer den samlede indtjening under hensyntagen til netværkets konkurrencedygtighed og kvalitet.

Bidragene i denne afhandling dækker modellering, metode og anvendelse. De udviklede metoder løser operationelle (rutning af containere), taktiske (hastighedsoptimering og ruteudvælgelse) og strategiske (netværksdesign) planlægningsproblemer, som containerrederier står over for. Ultimativt kan de foreslåede metoder hjælpe med at besvare spørgsmål som:

Hvordan kan kapacitetsudnyttelsen i netværket forbedres under hensyntagen til konkurrencedygtighed og kvalitet?

Hvordan skal nye ruter designes, så de bedst muligt gør nytte af eksisterende og nye markeder, og eventuelt forlader ikke-rentable markeder?

Hvilke ruter er det mest profitabelt at operere?

Hvordan skal ændringer i skibsflåden integreres i det eksisterende rutenetværk?

Hvilke skibe vil det være relevant at benytte i fremtiden?

Hvilke transporttider og antal omladninger vil det være relevant at tilbyde for en given transport?

# Preface

---

This thesis was carried out at the Division of Management Science, DTU Management Engineering, Technical University of Denmark as a partial fulfilment of the requirements for acquiring a Ph.D. in Engineering. Professor David Pisinger supervised the project and Professor Stefan Røpke acted as co-supervisor.

The thesis consists of three parts. The first part introduces the domain of liner shipping and show how large-scale strategic, tactical, and operational planning problems can be addressed. The second part develops algorithms for network optimization with transit time restrictions. The third part develops modelings and solution methods for service selection with limited transshipments. The thesis is based on eight academic papers that have been published or is currently under review in international peer-reviewed journals. They are each self contained with separate bibliographies. All papers are co-authored.

The project was supported by the Danish Maritime Fund under the “Competitive Liner Shipping Network Design” project and carried out in cooperation with Maersk Line, who has contributed to the discussion of the models and provided data such that the methods are tested under realistic conditions. The thesis was completed between December 15<sup>th</sup> 2012 and December 14<sup>th</sup> 2015.

Kongens Lyngby, December 14<sup>th</sup> 2015

Christian Vad Karsten





# Acknowledgements

---

Professor David Pisinger and Professor Stefan Røpke have been a truly amazing supervisor team! You have both supported my endeavors unambiguously and I am sincerely grateful to you for that. I deeply appreciate your genuine interest in my project and all the questions you have asked. In addition to this, Post. Doc. Berit Dangaard Brouer has been an invaluable collaborator and co-author.

Of importance for making my external stay at Massachusetts Institute of Technology so successful and inspiring were Professor Jim Orlin and Professor Andreas Schulz. I am very grateful that you gave me the opportunity to visit Sloan School of Management, where especially the Operation Research Center was a hub for many interesting ideas and fruitful discussions. I would like to give special thanks to Professor Anant Balakrishnan, who I met while at Sloan. I have truly enjoyed our collaboration during the last two years and it has given me another valuable perspective on modern optimization.

Special thanks also go to my colleagues at DTU Management Engineering, my co-author Professor Guy Desaulniers, and the Maersk Line vessel network optimization team, especially Mikkel Sigurd and Christian Plum. I have valued our collaboration and you have provided great feedback both as academic and industrial partners.

Finally, I am thankful to DTU for awarding me the PhD scholarship and to the Danish Maritime Fund who has supported the Competitive Liner Shipping Network Design project. Likewise, Skibsteknisk Selskab, Rudolph Als Fondet, Knud Højgaards Fond, Oticon Fonden, Henry og Mary Skovs Fond, Fabrikant Vilhelm Pedersen og Hustrus Mindelegat, Dansk Tennis Fond, and Laurits Andersen Fond have helped make my external research stay as well as international conference participation possible. These experiences will stay with me.



# Contents

---

Summary (English)	i
Resumé (Summary in Danish)	iii
Preface	v
Acknowledgements	vii
 <b>I Introduction</b>	 <b>1</b>
<b>1 Introduction and Motivation</b>	<b>3</b>
1.1 Level of Service . . . . .	5
1.1.1 Competitive Network Design . . . . .	6
1.1.2 Advanced Analytics Approach . . . . .	7
1.2 Maritime Logistics . . . . .	8
1.2.1 Industrial and Tramp Shipping . . . . .	9
1.3 Liner Shipping . . . . .	10
1.3.1 Industry Characteristics . . . . .	11
1.3.2 Operators and Operational Alliances . . . . .	16
1.3.3 Similar Networks . . . . .	18
1.4 Selected Planning Problems . . . . .	19
1.4.1 The Cargo Routing Problem . . . . .	20
1.4.2 Liner Shipping Network Design . . . . .	22
1.4.3 Maritime Speed Optimization . . . . .	23
1.5 Contributions . . . . .	26
1.5.1 Maritime Optimization . . . . .	26
1.5.2 Competitive Liner Shipping Network Design . . . . .	27
1.5.3 Service Selection with Limited Transshipments . . . . .	30
1.6 Conclusions . . . . .	33
1.7 Future Work . . . . .	35
1.7.1 Improved Modeling . . . . .	35
1.7.2 Additional Applications . . . . .	38

1.7.3	Extensions of Methods . . . . .	39
<b>2</b>	<b>Big Data Optimization in Maritime Logistics</b>	<b>51</b>
2.1	Introduction . . . . .	52
2.2	Liner Shipping Network Design . . . . .	55
2.2.1	Container Routing . . . . .	56
2.3	Matheuristic for Network Design . . . . .	59
2.4	Computational Results using <i>Liner-lib</i> . . . . .	63
2.5	Empty Container Repositioning . . . . .	64
2.5.1	Path Flow Formulation . . . . .	66
2.6	Container Vessel Stowage Plans . . . . .	67
2.6.1	Mathematical Model . . . . .	68
2.7	Bunker Purchasing . . . . .	70
2.7.1	Bunker Purchasing with Contracts . . . . .	71
2.8	The Vessel Schedule Recovery Problem . . . . .	74
2.8.1	Definitions . . . . .	74
2.8.2	Mathematical Model . . . . .	75
2.9	Conclusions and Future Challenges . . . . .	77
<b>II</b>	<b>Network Optimization with Transit Time Restrictions</b>	<b>83</b>
<b>3</b>	<b>The Time Constrained Multi-commodity Flow Problem</b>	<b>85</b>
3.1	Introduction . . . . .	86
3.2	The Level of Service in Liner Shipping . . . . .	88
3.3	Time-constrained Multi-commodity Flows . . . . .	90
3.3.1	Detailed Network Description . . . . .	94
3.4	Resource Constrained Shortest Path Calculations . . . . .	95
3.4.1	Reducing the Number of RCSP Calculations . . . . .	96
3.5	Graph Contraction . . . . .	97
3.6	Representation of Transshipments . . . . .	99
3.7	Computational Experiments . . . . .	104
3.7.1	Data . . . . .	104
3.7.2	Effect of Transit Time Limits and Implementation . . . . .	106
3.7.3	Effect of Transshipment Structure . . . . .	107
3.7.4	Effect of Reducing the Graph . . . . .	109
3.7.5	Sensitivity of Travel Time Restrictions . . . . .	110
3.8	Conclusions . . . . .	111

<b>4</b>	<b>Time Constrained Liner Shipping Network Design</b>	<b>119</b>
4.1	Introduction . . . . .	120
4.2	Mathematical Model for the TCLSNDP . . . . .	123
4.3	Algorithm . . . . .	126
4.3.1	Revenue Loss due to Transit Time Changes . . . . .	128
4.4	Computational Results . . . . .	130
4.4.1	Computational Results for <i>Liner-lib</i> . . . . .	131
4.5	Sensitivity . . . . .	134
4.6	Time and Transshipment Constrained Network Design . . . . .	135
4.7	Conclusions . . . . .	138
<b>5</b>	<b>Simultaneous Optimization of Sailing Speed and Container Routing with Transit Time Restrictions</b>	<b>141</b>
5.1	Introduction . . . . .	142
5.1.1	Industry Practice . . . . .	144
5.2	Transportation Network . . . . .	146
5.3	Mathematical Modeling . . . . .	148
5.4	Decomposition . . . . .	150
5.4.1	Decomposition and Solution of the MCF Problem . . . . .	153
5.4.2	Strengthening the Benders Cuts . . . . .	154
5.4.3	Generating Additional Benders Cuts . . . . .	156
5.4.4	Valid Inequalities . . . . .	156
5.5	Algorithm . . . . .	159
5.6	Computational Results . . . . .	160
5.6.1	Single Rotation Optimization . . . . .	161
5.6.2	Vessel Class Optimization . . . . .	163
5.6.3	Network Optimization . . . . .	167
5.7	Conclusions . . . . .	169
5.8	Appendix . . . . .	170
5.8.1	Optimal Time Scheduling . . . . .	170
5.8.2	Generating Additional Benders Cuts . . . . .	174
<b>6</b>	<b>Competitive Liner Shipping Network Design</b>	<b>179</b>
6.1	Introduction . . . . .	180
6.2	Problem Description . . . . .	184
6.3	Algorithm . . . . .	185
6.3.1	Level of Service Considerations . . . . .	187
6.3.2	Variable Speed on Service Legs . . . . .	192
6.4	Computational Results . . . . .	195
6.4.1	Computational Results for <i>Liner-lib</i> . . . . .	196
6.4.2	Sensitivity to Bunker Price . . . . .	199
6.5	Conclusions . . . . .	203
6.6	Appendix . . . . .	205

### III Service Selection with Limited Transshipments 211

<b>7</b>	<b>Container Shipping Service Selection and Cargo Routing with Limited Transshipments</b>	<b>213</b>
7.1	Introduction . . . . .	214
7.1.1	Background . . . . .	215
7.1.2	Planning Problems in Liner Shipping . . . . .	215
7.2	Literature Review . . . . .	217
7.2.1	Liner Shipping Network Design . . . . .	218
7.2.2	Related Problems . . . . .	219
7.3	LSP Problem: Definition and Model Formulation . . . . .	219
7.3.1	Problem Statement and Network Representation . . . . .	220
7.3.2	Model Formulation . . . . .	223
7.3.3	Problem Complexity . . . . .	226
7.3.4	Incorporating Trade and Operational Policies . . . . .	227
7.3.5	Problem Reduction . . . . .	230
7.3.6	Strengthening the Model . . . . .	231
7.3.7	LP-based Heuristic Solution . . . . .	233
7.4	Computational Results . . . . .	234
7.4.1	Test Problems . . . . .	234
7.4.2	Model Dimensions and Problem Reduction . . . . .	235
7.4.3	Results for the Base Model . . . . .	236
7.4.4	Effect of Strengthening the Model . . . . .	237
7.4.5	Heuristic Performance . . . . .	239
7.5	Conclusions . . . . .	240
<b>8</b>	<b>Optimal Selection of Liner Containership Services with Limited Transshipments</b>	<b>245</b>
8.1	Introduction . . . . .	246
8.2	Problem Definition and Model Formulation . . . . .	249
8.2.1	Problem Setting . . . . .	250
8.2.2	Modeling Approach . . . . .	252
8.2.3	Mathematical Model . . . . .	255
8.2.4	Operational Policies . . . . .	261
8.2.5	Complexity . . . . .	262
8.3	Model Improvements . . . . .	263
8.3.1	Strengthening the Model . . . . .	263
8.3.2	Forcing Constraints . . . . .	263
8.3.3	Cover Inequalities . . . . .	264
8.3.4	Heuristics . . . . .	266
8.4	Computational Results . . . . .	269
8.4.1	Model Dimensions and Problem Reduction . . . . .	269
8.4.2	Model Comparison . . . . .	270

---

8.4.3	Effect of Heuristics . . . . .	272
8.4.4	Sensitivity to Available Demand and Fleet . . . . .	273
8.4.5	Effect of Transshipment Limit . . . . .	275
8.5	Conclusions . . . . .	275
8.6	Appendix . . . . .	277





# Part I

## Introduction



## CHAPTER 1

# Introduction and Motivation

---

This thesis deals with “*the invisible network that keeps the world running*” (BBC, 2015b). It is comprised of numerous container vessels and is of a complexity that is impossible to comprehend for any human mind. It connects billions of consumers and producers across the globe and generates enormous amounts of data. This logistical data “*is distributed knowledge, managed by Maersk’s vast world-spanning computer network and shaped and interpreted by complex, similarly unknowable, algorithms.*” (BBC, 2015b). It is such algorithms we develop in this thesis.



**Figure 1.1:** Global Shipping Traffic. *Source: NCEAS (2008); Halpern et al. (2008).*

In 2015 a container volume corresponding to 170,000,000 twenty-foot containers is expected to be transported around the globe through this network (UNCTAD, 2014). The leading container shipping company transports more than 20,000,000 containers annually using a fleet of more than 600 vessels of varying size. The largest and most modern vessels can carry up to 20,000 containers, are more effective, and secures an environmentally friendly mode of transportation. However, these benefits are only fully realized in a network with high capacity utilization. This is challenging to achieve while utilizing the efficiency benefits and at the same time coordinating activities across the network in the best possible way. Advanced quantitative decision support tools will allow maritime transportation companies to operate at optimal profitability while sustaining or even enhancing the end-to-end experience.

The algorithms developed in this thesis are intended as decision support tools for liner shipping companies to optimize their networks while both *carrier* (a specific liner shipping company) and *shipper* (the customer who owns the cargo) requirements are integrated to maximize quality and profitability. The tools impact the planning process in two principal ways. Firstly, a carrier will quickly be able to assess the impact of their decisions and get an overview of the ripple effects across the network. Secondly, the algorithms can provide insights on the design of new routes and optimization of existing. They can suggest the best improvements to one or several sailing routes while still considering the effects across the entire network. These improvements can both be in terms of better container routings, changing sailing speed, or including new ports, routes, or even new markets. Likewise, they can help determine if it is better for a service no longer to visit a given port or market. From a practical perspective a tool to support incremental changes and improvements is of great value. This way it can assist managers and planners to quickly adapt to new situations including macro economic changes, seasonal effects, holidays, environmental restrictions, bunker price variations, negotiation of alliances, fleet renewal etc. The quantitative tools developed in this thesis can help answer questions that are of a complexity where companies previously have had to rely on gut feeling and qualified guesstimates. Related to the physical layout of the network, one of the core decisions in the network design process that carriers must choose is the travel times offered and whether to deliver containers using direct connections or transship at an interim hub. These decisions depend among others on the vessel sizes, freight rates, demand volumes, and distances and the trade-offs are constantly changing as fuel prices are fluctuating and vessels become larger. The tools can help understand how changes in the fleet can best be integrated into the existing network, what ships will be relevant to use in the future, and how the capacity utilization of a network can be improved while considering customer requirements. Furthermore, the tools will be essential to a company

Customer Perspective	Carrier Perspective
Primary	
Low cost	Cost effectiveness
Short transit time	Sufficient capacity
High reliability	Operational alliances
Few transshipments	
Secondary	
Global coverage	Robust operation
Drayage (inland transportation)	Strategic goals
Specific departure days	Comply with legislation
Corporate social responsibility	
Equipment and administration	

**Table 1.1:** Level of service requirements driving the design of a liner shipping network. *Source: Maersk Line and Brouer et al. (2014a).*

when cooperation agreements are negotiated. For a global network it is impossible to assess the adjustments necessary to maintain a competitive network without using advanced planning tools. The proposed methods support the decision process while taking into account the competitiveness and quality of the network both from a carrier and customer perspective.

## 1.1 Level of Service

The competitiveness of a liner shipping network is determined by the *level of service* requirements that are considered while designing the network. The level of service offered to the shipper is defined by several aspects that all influence the network design. Some of these are summarized in Table 1.1 and discussed further below. The most important aspects to consider for the *product* offered to the shipper is the combination of cost and transit time. This is the focus of **Part II** of this thesis **Network Optimization with Transit Time Restrictions** where we develop algorithms for liner shipping network optimization with transit time restrictions. The number of transshipments is also a crucial parameter and this will be discussed briefly in **Part II** and addressed further in **Part III: Service Selection with Limited Transshipments** where we develop models and solution methods for service selection with limited transshipments.

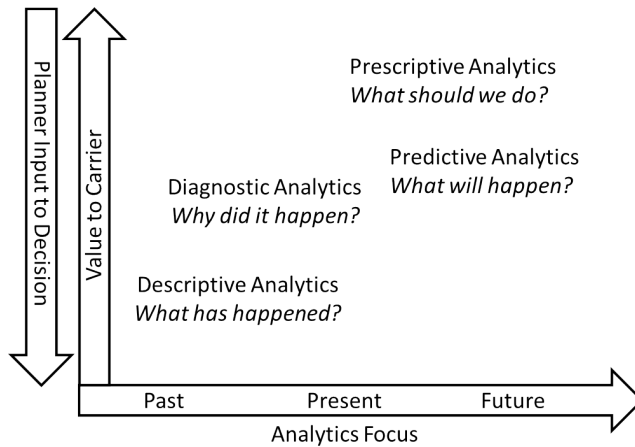
Containers are transported through the network from port A to port B, and a *transport* may include the use of several services to connect between the

origin and the destination port. We refer to the transits between services as *transshipments* and the *transit time* is the time used to transport a container from origin to destination. In practice, transit times vary from one day to several months and most containers are transshipped no more than two times, although some containers can be subject to up to five or six transshipments. The number of transshipments can vary based on the origin and destination region. Generally, intra-region cargo has fewer transshipments than inter-region shipments, as might be expected, but the structure also varies across different regions and is not symmetric, i.e., it can vary depending on the direction of shipment between two regions as will be shown in Chapter 8. A network with direct connections between all serviced ports would offer low transit times and no transshipments, but at the same time it would be very expensive to operate for the carrier as most pairs of ports do not have enough container demand to fill a vessel. This illustrates the inherent trade-off between the cost of networks versus transit time and/or number of transshipments.

The transshipment operations require resources at intermediate ports to unload, store, and re-load the container. Therefore, they are both expensive and time consuming. From a customer's perspective, having fewer transshipments generally reduces handling time, possibly transit time, the risk of damage, and also the risk of missing connections. Therefore, transportations which entail fewer intermediate transshipments are preferred. Moreover, customers may specify the maximum number of permitted transshipments for their containers in order to limit handling, damage, delays, and layover times, especially for hazardous and high-value cargo. On the other hand, transshipment operations are important for carriers since they permit using vessel capacities more effectively. However, from a carrier perspective having fewer transshipments reduce the non-value adding steps, handling capacity requirements, and transshipment costs (Balakrishnan and Karsten, 2015b). The practice of limiting the number of transshipments will be evident from data based on the operation of a global carrier presented in **Part III**. Transit time and number of transshipments are correlated, and generally limiting one of them will implicitly bound the other, e.g. a container that has a transit time requirement of one week can only transship very few times before the allowed transit time is violated simply due to handling and layover time. Likewise, it is not possible to reach a certain part of the network if a customer allows only one transshipment, which implicitly puts a bound on how long a transit time can be.

### 1.1.1 Competitive Network Design

A competitive network design can broadly be defined according to the perspectives included during the design process:



**Figure 1.2:** Business value of analytics models versus temporal focus and difficulty. *Inspired by: Puget (2015).*

For a liner shipping network to be competitive it must be:  
*optimized in terms of a traditional economic criteria from the point of view of the carrier while respecting the view of the shipper by including level of service requirements in order to attain a balance between the two dual perspectives.*

Being able to construct routes satisfying a given level of service makes it possible for a liner shipping company to request higher transportation costs from the customer, differentiate customers according to level of service, and possibly even attract new customers. However, the operation of global carriers is of a scale where humans can not comprehend and leverage the full potential, so to be competitive, carriers need to do more than just collecting and storing information.

### 1.1.2 Advanced Analytics Approach

The way modern organizations operate is changing and enterprises are discovering how Big Data from operations and the use of advanced analytics can lead to new types of insights revolutionizing business processes. There is evidence showing that data-driven decisions tend to be better decisions (McAfee and Brynjolfsson, 2012; Brynjolfsson et al., 2011). Therefore, to optimize business processes competitive strategies need to be built around data-driven decision support tools (Davenport and Harris, 2007). Figure 1.2 illustrates one view



of the different insights that can be obtained for different types of analytics models used in the process related to finding the best configuration of the network. The chart has model focus (time) as one dimension and value to a carrier based on progress towards a decision as the other. As described by Puget (2015), the value of the tools will depend on to what degree the decision process is automated and thus how much (or little) input is required from the planners in the network decision process. Descriptive and diagnostic models are concerned mainly with simple statistics on past performance, i.e. what happened and why did it happen. Predictive models are more advanced and usually rooted in machine learning and can help to predict what will happen in the (near) future. However, only in simple cases they directly suggest an action. Prescriptive models are usually optimization based. They are the focus of this thesis and their aim is to suggest actions and provide decision support or even decision automation on what the best action will be.

Liner shipping companies operate under changing economic conditions, and predictive analytics may provide improved input for the prescriptive solutions provided by optimization to obtain more robust decisions as discussed in Chapter 2. Furthermore, leading carriers are investing in devices and software to track containers in real time and emerging *Internet of Things* technologies (Porter and Heppelmann, 2015) and Big Data analytics methods (Davenport, 2014) do enable transportation networks to “think” by themselves to a larger degree than what has previously been possible. It is unlikely that carriers will reach a point where all decisions will be automated completely and purely based on data. There will still be many decisions that can and should not be data-driven such as one-time events that have never happened and part of the strategy in relation to competitors, alliances etc. However, by using analytics models it is possible for carriers to turn data into actionable insights and provide them with abilities that they have not had in the past. From a network design perspective it is possible to e.g. identify under-utilized and poor-performing resources and automatically propose actions to improve the current network.

## 1.2 Maritime Logistics

Comprehensive decadal reviews of operations research (OR) research published in scientific journals and edited volumes on ship routing and scheduling are provided by Ronen (1983, 1993); Christiansen et al. (2004, 2013). Christiansen et al. (2007) give an excellent introduction to the use of OR within maritime transportation and Hoff et al. (2010) survey the literature related to combined

fleet composition and routing in maritime (and road) transportation and describe some industrial aspects. The first maritime applications in OR dates back to the 1950's and concern naval tankers (Dantzig and Fulkerson, 1954; Flood, 1954), while much research today is focused on industrial applications, especially liner shipping (Christiansen et al., 2013).

The field of maritime logistics is rather broad and includes problems within both industrial, tramp, and liner shipping. The operating characteristics of the three are rather different and decision support tools from one area rarely applies directly to another area. However, as will be seen e.g. in Chapter 6, some ideas from tramp shipping can be transferred to liner shipping and therefore we will give a short introduction to the three areas with emphasis on liner shipping as this is the focus of this thesis.

Other aspects related to maritime logistics such as terminal operations and intermodal transportation are also presented in the OR literature. Vis and De Koster (2003); Stahlbock and Voß (2008) present an overview of planning problems that arise at container terminals, Bierwirth and Meisel (2010) give an overview of berth allocation and quay crane scheduling, and Crainic et al. (2006) discuss the land-side operations of terminals.

There is also a stream of literature on inventory routing problems (IRP) (Coelho et al., 2013) where many of the applications are seen in maritime logistics, e.g. the liquefied natural gas (LNG) IRP (Grønhaug and Christiansen, 2009). More generally in Maritime Inventory Problems (MIR), ship routing, delivery scheduling and inventory management are integrated with the responsibility for the inventory management and for the ships' routing and scheduling being handled by the same actor (Christiansen and Fagerholt, 2009; Andersson et al., 2010; Christiansen et al., 2013).

Recently, benchmark data sets resembling real world maritime operations of industrial shipping, tramp shipping, liner shipping, and MIR have been published, which can help mature the research area (Brouer et al., 2014a; Hemmati et al., 2014; Papageorgiou et al., 2014).

### 1.2.1 Industrial and Tramp Shipping

Usually, shipping is characterized according to three general modes of operation, namely industrial, tramp, and liner (Ronen, 1983, 1993; Christiansen et al., 2004, 2013). The methods developed in this thesis are all concerned with liner shipping, but some methods and considerations apply to all three modes. In Chapter 5 we draw on research from tramp shipping to develop our speed optimization model. Likewise, we believe some of the results from this thesis related to optimization with level of service considerations can be generalized

to tramp and industrial shipping. Here we briefly describe the characteristics of tramp and industrial shipping. Further background and an overview of the more recent literature can be found in Christiansen et al. (2013). In industrial shipping, the shippers also controls the ships and their goal is to ship all their cargo at minimal cost. In most work on routing in industrial shipping the objective is to minimize the cost of a fixed fleet of (heterogeneous) ships while transporting one or several bulk products. In tramp shipping, companies usually have a certain amount of cargo that they are obligated to carry. The goal is then to maximize the profit from optional cargoes. Here the operator normally controls a heterogeneous fleet of ships that are available to transport the cargo (the operation is different from liner shipping where companies operate according to a published itinerary and schedule and all cargo is optional). A cargo in industrial and tramp shipping consists of a certain amount of one or several products to be picked up at a specified origin port, and unloaded at a specified destination port. Industrial and tramp shipping deals with bulk cargoes that are shipped in large quantities, such as oil, coal, iron, and chemicals. These are usually shipped in full shiploads from their origin port to their destination port. Sometimes minor cargoes are also transported which may require routes with multiple stops. This has lead to research on both full shiploads and multi-stop routes. There is usually a time window associated with a given cargo during which it must be loaded and unloaded. Tramp shipping can be seen as a generalized version of industrial shipping and it includes most of the characteristics from here. The core planning problem facing a tramp shipping company is to select spot cargoes and construct routes and schedules that maximize profit. From a mathematical perspective this has many similarities with the multi-vehicle pickup and delivery problem with time windows (Desrosiers et al., 1995)

### 1.3 Liner Shipping

Liner shipping involves transporting goods by means of high-capacity ships that service regular routes following fixed schedules. Liner ships are mainly container ships and throughout this thesis liner and container ship will be used interchangeably. In the sequel we will describe the most important characteristics of the industry along with the most commonly used industry terms. The container liner shipping domain is thoroughly described in Stopford (2009); Alderton (2011); Brouer et al. (2014a). Meng et al. (2014) review the OR literature focusing on liner shipping. A poetic ode to the container industry for the layman can be found in De Botton (2010), and Levinson (2010) gives a historic overview of how container shipping developed into the industry that

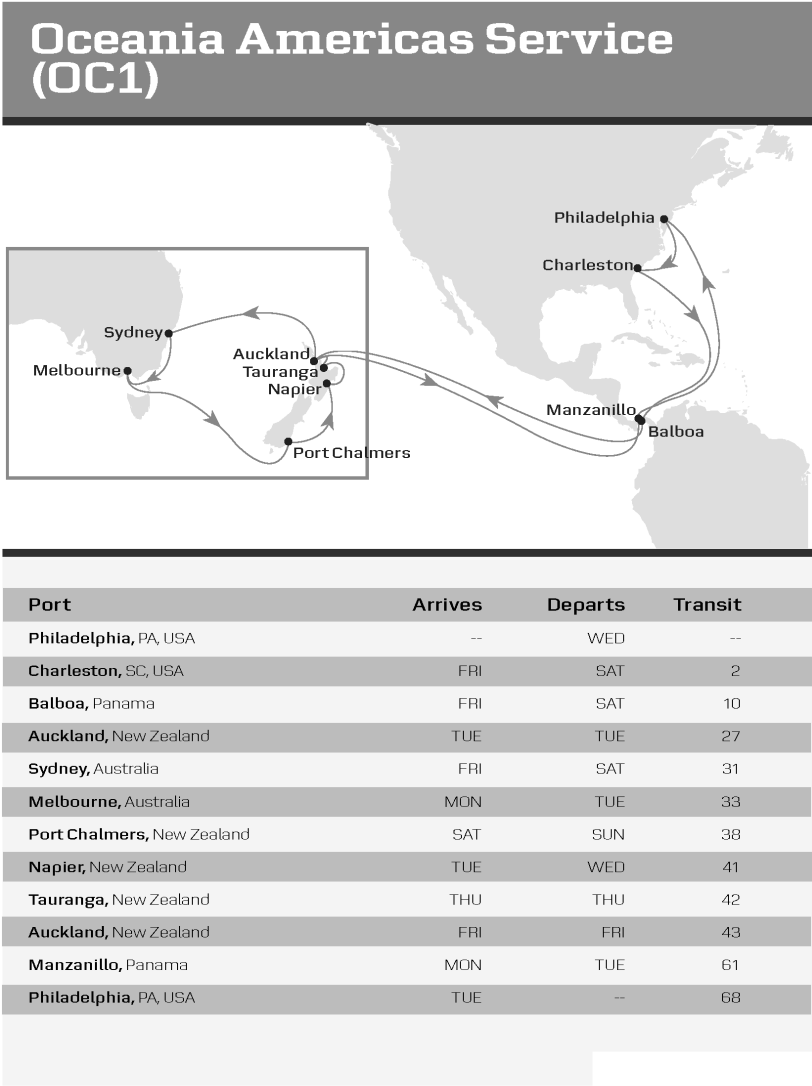
today makes global trade possible. Here we will go through the most important details related to the operation of liner shipping networks.

### 1.3.1 Industry Characteristics

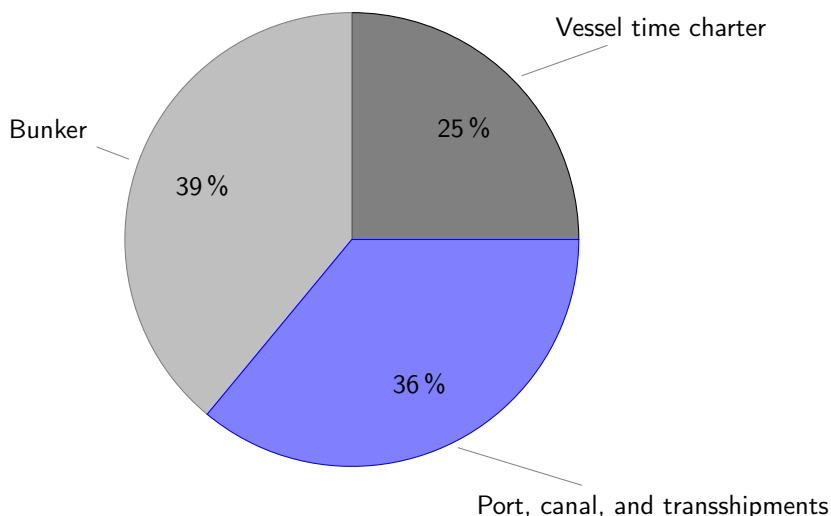
In liner shipping a *service* is a round trip sailed at a fixed frequency. During a round trip a set of ports is visited and the timing of events along the service is given by a *schedule*. Figure 1.3 shows an example of service operated by Maersk Line between Oceania and the Americas. The total round trip time is 10 weeks and for each port the arrival and departure day is given together with the total travel time. The Asia-Europe route, via the Suez Canal, is one of Maersk Line's busiest and is responsible for around one quarter of Maersk Line's business. Usually services can be divided into a head and a back haul, where the head haul is usually the cargo intensive part, e.g. from Asia to Europe. The number of ports visited by a service varies from a few to more than 20. *Trunk services* serve main ports with several demands across several regions and *feeder services* are usually used within a specific market to serve a single main port and a set of smaller feeder ports. A *fleet* of vessels is deployed to the services such that the capacity and speed is in accordance with the demand maintaining the desired frequency. Global carriers generally deploy vessels with similar characteristics to a service to reduce the complexity of the network design and corresponding schedules, (Notteboom and Vernimmen, 2009; Stopford, 2009). A network is usually divided into *trades*, e.g., Asia to Europe or Asia to North America. Each trade has special characteristics and each service will normally focus on individual trades or in some cases several trades.

The major cost components of operating a network are associated with the fleet and cargo handling costs. Figure 1.4 shows the breakdown of the major cost components of a network operated by a global carrier. The main costs associated with the fleet include bunker cost (the fuel consumed by container vessels), port and canal charges (there is a fee for calling a port and traversing canals, both are vessel size dependent), and financing of vessels (this includes capital costs of acquiring or financing a vessel and the operational cost (OPEX) which includes crew, maintenance, and insurance). Another previous breakdown can be found in Stopford (2009) who estimated the bunker cost to be 35-50 % of a vessel's operational cost, capital cost to be 30-45 %, OPEX to be 6-17 %, and port cost to be 9-14 %.

The cargo handling cost is calculated from the load and unload cost at the origin and destination port and the cost associated with transshipments at intermediate ports. In addition to this there are costs for the shipper associated



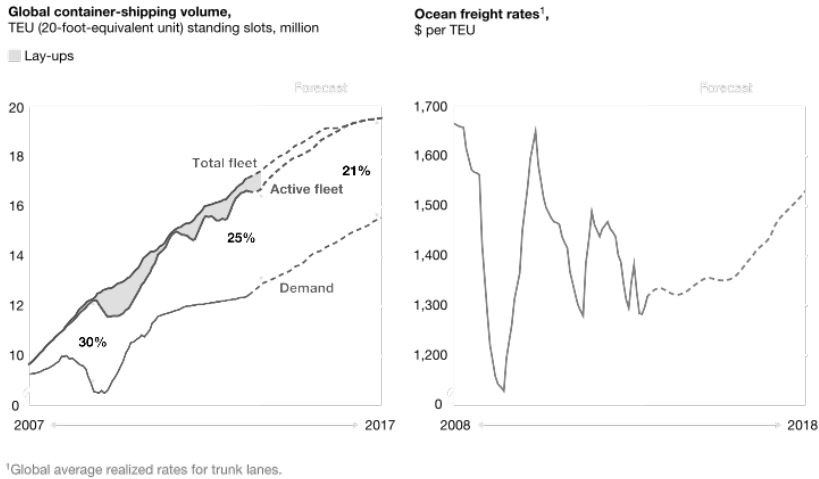
**Figure 1.3:** Service operated by Maersk Line. Itinerary, corresponding schedule, and transit time from start port are given. The service is a so-called *butterfly* where multiple connected cycles are centered around one port (Auckland), which is visited twice during one round trip. *Source: Maersk Line.*



**Figure 1.4:** 2015 cost breakdown of the Maersk Line network. *Source: Maersk Line.*

with owning or leasing containers. The load and unload costs do not depend on the routing of the container, whereas the transshipment cost does, and hence on the total number of transshipments. As will be seen in Chapter 8 a global carrier will not provide direct connections for a significant percentage of the available cargo but most cargo will be transported using up to two transshipments. In practice the cargo handling cost can have volume dependent costs at some terminals.

Revenues are obtained by transporting cargo but as seen in Figure 1.5 the freight rates are highly volatile whereas the trend in demand is increasing, except just after the 2008 financial crisis, and this increasing trend is expected to continue. Between 1983 and 2006 the growth in container cargo was 10% on average per year, Stopford (2009). The demand, supply, and bunker price are the main determinants for the freight rate offered to a shipper to transport a specific cargo. In addition to this it depends on the transit time, the container type needed e.g. refrigerated containers are more expensive, special regulations related to restricted and dangerous goods, the number of transshipments, and the nationality (flag) of the vessel. As the inventory costs are paid by the shipper this also highlights why transit time is a crucial factor. Furthermore, there are large imbalances in world trade while the supply between different regions is symmetrical due to the nature of the routes. Hence, transporting a container from Asia to Europe can easily cost three times more than the reverse (Brouer et al., 2014a). In 2014 11.5 mio. twenty-foot equivalent units



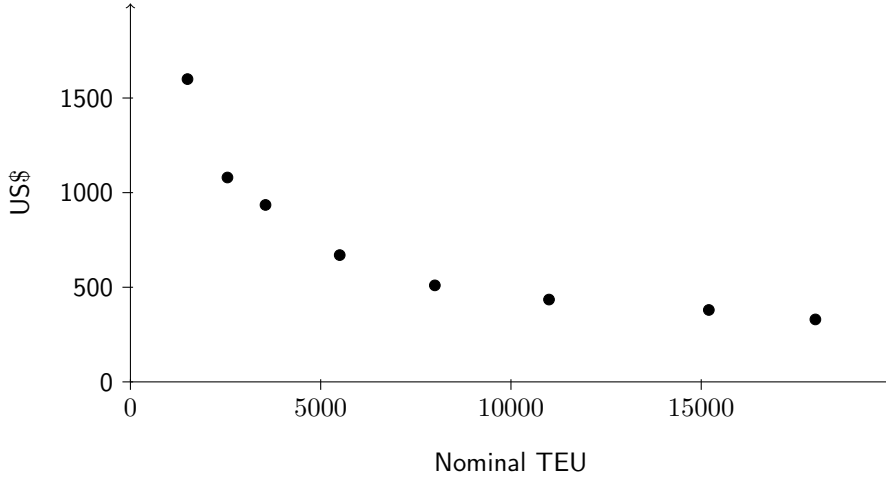
**Figure 1.5:** Demand trend and freight rates *Source: Glave et al. (2014).*

(TEU) were transported from Asia to Europe while only 5.5 mio TEU were transported from Europe to Asia (UNCTAD, 2014).

The seasonality in the production of some products over the year affects both the global demand levels but more specifically it affects single ports or trades. One peculiarity of container shipping is that the shipper will most often only pay for a container transport once it is executed since there is no fee for booking cargo. This leads to many issues for the carrier in terms of forecasting demands and most carriers will overbook departures because of no-shows.

Generally, because of economies of scale, larger vessels are cheaper to operate per TEU. The market rate of a vessel is referred to as the *Time Charter Rate* (TC rate) and corresponds to the cost of chartering (lease) in a container vessel to the existing fleet or charter out an owned vessel to another carrier. TC rates are highly dependent on the length of the chartering period, seasonality, and general economic environment. Often carriers own a core fleet but supplement it by chartering in and out vessels to meet changing capacity requirements and thereby gain flexibility. TC rates include OPEX, capital cost and depreciation of the vessel's value and the capital cost and OPEX varies with capacity (Brouer et al., 2014a).

However, even though economies of scale make larger vessels cheaper to operate per TEU, a limiting factor for increasing the vessel size is that the time spent in port increases to a relatively large part of the journey (OECD, 2015). This is among other because of a legacy in the design of quayside container gantry



**Figure 1.6:** Estimated annual operation cost per nominal TEU – assuming 85 % utilization. *Own estimates based on Figure 2.2 in OECD (2015).*

cranes that are based on smaller vessels. The cranes are mainly manually operated and can only carry one container at a time. Automated operation and new designs that can carry several containers may increase port productivity enough to make it feasible to design even larger vessels than operated today, but at the current port productivities this is less attractive. Additionally, larger vessels increase the supply chain risks and they would require a substantial extension of the general infrastructure which may be more costly than the potential savings as there currently seems to be diminishing returns. Figure 1.6 shows estimated annual operation cost per TEU as a function of vessel size. Most of the savings for the newer ships are because of more efficient engines and not because of scale (OECD, 2015).

More details on cost and revenue structure can be found in Stopford (2009); Brouer et al. (2014a); OECD (2015).

In the airline industry the sale of the very large four-engined jumbo have declined recently and they have been replaced by smaller more flexible planes (BBC, 2015a). The same may happen in the liner shipping industry where smaller carriers operating only at the most profitable strings may be a serious competitor. Furthermore, as noted by Krugman (2013) “*ever-growing trade relative to GDP isn’t a natural law*”. In fact, recently the global capacity has been increasing faster than demands which has resulted in the largest and most modern vessels being idle. This is mainly because that if you use the largest ships while demand is decreasing, the only way to fill them is to sail less fre-





**Figure 1.7:** Grams of  $CO_2$  emitted by transporting 1 tonne of cargo 1 km. *Source: (Maersk, 2015).*

quently, and then the customers' expectation will not be satisfied (WSJ, 2015; Ship&Bunker, 2015).

The environmental impact of a global liner shipping network is significant, but compared to other modes of transportation container shipping offers a greener alternative as illustrated by Figure 1.7. During the past decades a lot of research has gone into reducing the fuel consumption of vessels and thus the environmental impact. In addition to this a well-designed route net potentially comes at very low cost but with a potential large benefit as it will not only improve the competitiveness but also the sustainability of shipping.

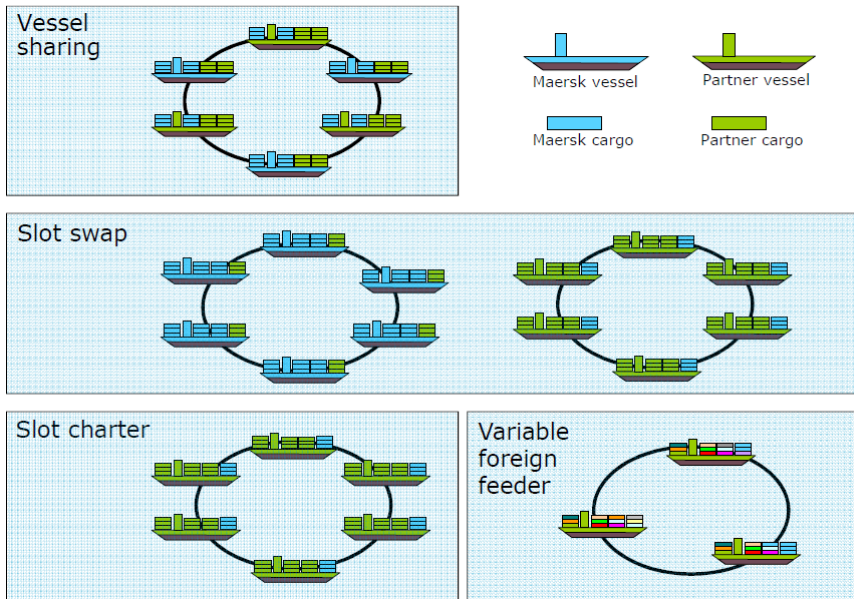
### 1.3.2 Operators and Operational Alliances

Name	Carriers
<i>2M</i>	Maersk Line and Mediterranean Shipping Company (MSC)
<i>Ocean Three</i>	China Shipping Container Lines Company, CMA CGM, and United Arab Shipping Company
<i>CKYHE</i>	China Ocean Shipping (Group) Company (COSCO), "K" Line Ship Management Company, Yang Ming Group, Hanjin Shipping Company, and Evergreen Marine Corporation
<i>G6</i>	APL, Hapag-Lloyd, Hyundai Merchant Marine Company, Mitsui O.S.K. Lines, Orient Overseas Container Line, and NYK Line

**Table 1.2:** Operational Alliances.

As of 2015 the major global carriers are allied in four large groups, summarized in Table 1.2. These alliances allow carriers to leverage the economies of scale of operating larger vessels, and especially for smaller carriers the alliances help them to be able to offer a global network without making large investments

in increasing their fleet. As shown in Table 1.1, operational alliances are one of the main drivers of the network design. Figure 1.8 shows the main types of different partnerships. *Vessel sharing agreements* (VSA) entails that two or more carriers share a service such that they can deploy larger vessels and provide higher frequency. Each carrier will operate a share of the vessels and have a corresponding share of the capacity on all vessels on the service. VSA are complicated to initiate because each partner rarely operate matching services and have identical available vessels. However, today VSA are a central part of liner shipping (Brouer et al., 2014a). VSA are studied in a game theoretic framework in connection with cargo flow and network design in Agarwal and Ergun (2008a, 2010). *Slot Swap Agreements* (SSA) are entered by two carriers who enter a contract to use slots (capacity) on each others services or parts of them. *Slot Charter Agreements* (SCA) are similar but here it is simply one carrier who enters a contract to use slots on another carrier's service or part of it. Finally, the last main type of partnerships are *Foreign Feeder Agreements* (FEF), which are entered with feeder carrier who will combine volumes from many carriers. A service operating between a central hub port and a few minor ports is called a feeder and is usually operated by smaller vessels.



**Figure 1.8:** Partnering at Maersk Line. *Source: Maersk Line.*

The operational alliances are in effect for the part of the supply chain at sea and significant economies of scale are achieved. However, on the land side of

the operation each carrier has its own agreements related to hinterland transportation, so here the lines do not leverage the scale advantage (Joeress et al., 2015).

### 1.3.3 Similar Networks

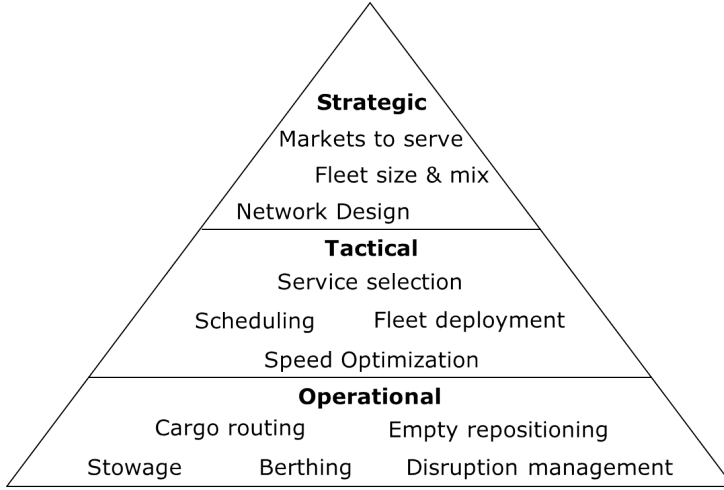
Public transportation networks show some of the same characteristics as liner shipping networks. The integrated line planning and passenger routing problem from public transport determines which lines to operate in a public transportation network. This is usually done while also handling the duality of the passenger who wants shorter travel times with few transfers and the public transportation company who wants lower costs. This is a similar trade-off as the one seen in liner shipping between cost/profit on one side, and the container travel time and number of transfers on the other. As in liner shipping the balance of the two significantly impact the layout of a public transportation network (Schöbel, 2012; Schmidt, 2014). It has been shown that it is important to integrate routing decisions in many of the public transportation planning problems when optimizing both cost and transit time as these are highly dependent problems (Schmidt and Schöbel, 2015). In railway optimization the planning problems include both routing of passengers and cargo through a network. The railway line planning problem is the problem of designing a line system such that all travel demands are satisfied while optimizing some objective. Also here the two main conflicting objectives to be optimized are again similar to liner shipping. It is desired to maximize the service towards the passengers and minimize the operational costs of the railway system (Caprara et al., 2007). However, where the lines in public transportation are constrained by roads or tracks, liner shipping seldom have such constraints.

A commonly studied problem in public transportation is the line planning problem (Schöbel, 2012). This problem entails selecting a number of lines from a candidate pool (potentially containing all possible lines) which provides sufficient passenger capacity and meets various operational requirements, while optimizing an objective related to the quality of a line plan. However, in public transportation all passenger demands are usually modeled such that they must be met, whereas in liner shipping demands are optional and only profitable cargoes are selected. So where the objective of public transportation networks involves cost or travel time minimization liner shipping models are formulated in terms of profit maximization. Borndörfer et al. (2007) present a multi-commodity flow model that generates passenger and line paths dynamically, which conceptually is similar to generating container paths and sailing routes. The pricing of passenger variables is a shortest path problem and the

pricing of line variables is solved as a (length restricted) maximum weighted path problem. A two phase algorithm is used where the first phase consist of running the column generation algorithm while reduced cost passenger paths or line paths are found. Then the second phase solves an integer program using the obtained line pool of non-zero frequency in the LP solution. Only simple lines are found and more importantly, passenger transfers between lines are not accounted for. Later work by Borndörfer and Karbstein (2012) introduce a transfer estimate on a given path to estimate the number of passenger transfers. The model leads to an approximation in terms of transfers and underestimates the number of transfers on a path. The following idea is used to estimate the number of transfers. The capacity constraints related to direct connections bound the number of direct passengers on a specific arc by the capacity of the lines that connect origin and destination. Hence, some passenger paths must use at least one transfer if this capacity is exceeded. However, some capacity of the lines serving a specific arc may be used up by other passengers such that capacity constraints overestimates the capacity for direct passengers. Therefore, the model can in the worst case calculate zero transfer estimates on passenger paths that actually require at least one transfer. However, on realistic test instances the model turns out to estimate the number of transfers quite accurately. To calculate the number of transfers correctly a “change-and-go” model is also derived but this approach only works for small, explicitly computable line pools. The graph of the “change-and-go” model has for every line a copy of each node and edge and transfer edges are added. Hence this graph has similarities with the one used in this thesis. However, the network design characteristics are different as public transportation lines are not cyclic in the way container shipping routes are, but they all have frequency requirements to be able to offer regular service.

## 1.4 Selected Planning Problems

Chapter 2 of the **Introduction** will present a selection of the most important planning problems in liner shipping and show how strategic, tactical and operational problems can be addressed using large-scale optimization techniques. Figure 1.9 shows some of the main planning problems encountered in liner shipping. Among the most important strategic decisions are the decisions of which markets to serve, the fleet size, the composition of the fleet (mix), and the actual design of the sailing routes (network design). The tactical problems include a selection of which services to operate and the fleet deployment, sailing speed, and scheduling of these. Key operational decisions are the routing of cargo through the network and repositioning of empty containers, but



**Figure 1.9:** Main planning problems in liner shipping.

also berthing of vessels in ports, the stowage of containers on the vessels, and disruption management are important decisions for a network to be able to operate in the best possible way. In the OR literature the problems are often treated somewhat independently (Christiansen et al., 2013; Meng et al., 2014), but as will be seen throughout this thesis the problems are often highly dependent, e.g. the best design of a set of services depends very much on the sailing speed on each leg, which again influence the level of service a customer receives thus the optimal cargo routing and vice versa. However, solving each of the problems independently is challenging enough and valuable insights can be obtained by considering each problem, or a subset of these, independently. The planning problems can also be classified according to different operational conditions (Kjeldsen, 2011). In the following we briefly discuss the problems of cargo routing, network design and speed optimization as these are central to this thesis. For a comprehensive review of the OR literature focusing on liner shipping planning problems and a characterization of these we refer to Meng et al. (2014).

### 1.4.1 The Cargo Routing Problem

At the heart of many of the planning problems in liner shipping lies the problem of cargo routing. Mathematically, the problem can be formulated as an extension of the multi-commodity flow problem (Ahuja et al., 1993). It comes

in an arc-flow formulation and in a path flow version with exponentially many variables. However, in practice it turns out that the path-flow formulation is often more attractive to solve using column generation such that all variables are only considered implicitly (Desrosiers and Lübbecke, 2005; Lübbecke and Desrosiers, 2005). Brouer et al. (2011) solve the multi-commodity flow problem with additional inter-balancing constraints to control repositioning of empty containers using a column generation approach. They benchmark against solving the arc-flow formulation and find that it is advantageous to solve the path flow formulation for the liner shipping application. In theory, an integer version of the multi-commodity flow problem should be solved as containers can not be split. Integer solution can be obtained e.g. using a branch-and-cut-and-price approach (Barnhart et al., 2000), but this makes the solution process much less attractive as a sub-problem in network design algorithms. Fortunately, thousands of containers are transported between different origins and destinations, so fractional solutions may be of less concern and demands are not known exactly. In fact, Brouer et al. (2011) show that obtaining integer solutions to the cargo routing problem by rounding down the LP-solution provides extremely good quality solutions that are within 0.01 % of the upper bound of the linear relaxation.

The cargo routing problem and the solution of the associated multi-commodity flow problem with level of service considerations is the focus of Chapter 3. Chapter 3 - 8 all contain different variations of the cargo routing problem and the majority of the work related to network design presented in the previous section and several tactical level planning problems solves a multi-commodity flow problem to evaluate flow in a network (Fagerholt, 1999; Shintani et al., 2007; Agarwal and Ergun, 2008b; Wang and Meng, 2012; Brouer et al., 2014b; Plum et al., 2014). In Chapter 3 - 6 a path flow formulation is used while arc flow formulations on augmented networks are used in Chapter 7 and 8. If the integer part related to the service selection in the problems considered in Chapter 7 and 8 are fixed, the model corresponds to solving the linear version of the arc-flow formulation of the multicommodity flow problem with limits on the number of transshipments. It turns out that the computational solution times for the arc flow formulation of the transshipment constrained multi-commodity flow problem using an augmented network are comparable to the solution times for the time constrained multi-commodity flow problem solved using column generation.

### 1.4.2 Liner Shipping Network Design

Container ships usually operate along established routes following regular time tables published several months ahead. The routes may not change significantly for years although the frequency and time tables may change more often. Therefore, the design of the routes is one of the most important strategic decisions. In practice, routes are also often operated in cooperation with other companies as described in Section 1.3.2 complicating the design process further. The problem facing liner shipping companies of constructing routes and choosing which routes to serve is usually referred to as the *network design problem*. The core problem associated with liner shipping network design is to determine which ports the container ships should visit and in which order. Furthermore, the frequency of the routes must be determined along with the size and speed of the ships used. Mathematically the liner shipping network design problem is strongly NP-hard (Brouer et al., 2014a) and is related to the capacitated network design problem (Magnanti and Wong, 1984; Balakrishnan et al., 1997; Gendron et al., 1999; Crainic, 2000) and the vehicle routing problem (Laporte, 2009; Toth and Vigo, 2014). In the latest of the decadal reviews Christiansen et al. (2013) identifies the liner shipping network design problem as the single most important problem to be properly addressed within maritime transportation. Christiansen et al. (2007) also emphasize the need for quantitative decision support systems.

The most general version of the problem involves many ports and services in the network, does not impose a specific network structure such as hub-and-spoke, and allows for container transshipment operations. Agarwal and Ergun (2008b) present one of the first scalable approaches to this problem. Furthermore, they are the first to include transshipments. The model is formulated for a time-space graph and creates routings for a set of vessel classes. It is one of the only network design models to also include a rough schedule. However, the formulation does not model transshipment costs correctly and does not include travel time considerations in the container routing though using a time-space graph. Álvarez (2009) formulates a mixed integer model with a more correct transshipment cost representation, but still the model has issues with transshipment costs when including more complex routes, e.g. butterflies as pointed out by Brouer et al. (2014a). The model reduces to a linear multi commodity flow problem when the integer variables related to vessel routing are fixed. A tabu search is devised to move between different integer solutions where routes are generated by column generation and the flow problem is solved by an interior point method. The solution of the MCF problem is identified as a general bottleneck in local search methods as excessive time is spent on the solution of this. Brouer et al. (2014a) extend the work of Álvarez (2009) and propose

a model where transshipment costs are correctly accounted for but as Álvarez (2009) also solves the problem using tabu search, which only explores a limited part of the solution space for instances of realistic size. Brouer et al. (2014b) propose a new solution method for the model proposed by Brouer et al. (2014a). Based on an initial set of routes an improvement (mat)heuristic based on the solution of a mixed integer program is proposed. To date the most successful approaches have been heuristic, but some attempts have been made at creating exact methods for the problem. Reinhardt and Pisinger (2012) propose the first branch-and-cut approach to the problem that considers transshipment costs. The method is able to create simple as well as non-simple butterfly routes. Results are reported for instances with up to 15 ports.

Early work on network design did not consider transshipments because the importance of these used to be much less significant from an industry perspective. Therefore early formulations and solution methods for the network design problem did not include transshipments (Rana and Vickson, 1991; Fagerholt, 1999). The model by Rana and Vickson (1991) was later extended by Shintani et al. (2007). The extension relaxed a restrictive visiting order to represent a more realistic set of routes and repositioning of empty containers is also included but transshipments are still not considered. To solve the problem a genetic algorithm is used. Later, the importance of transshipments in modern liner shipping networks has been emphasized (Baird, 2006; Notteboom and Rodrigue, 2008; Brouer et al., 2014a). Other contributions consider less general versions of the problem, e.g. in a version with only a single route or a version with a set of routes but no transshipments allowed. Other approaches identify some ports as hub ports and feeder ports are assigned hubs where they can transship a priori. For a review on literature on the more restricted problems we refer to Christiansen et al. (2013).

Gelareh et al. (2010) have a more game theoretical approach to network design and investigate the competitive position of a carrier in a market where a newcomer carrier is entering. They propose a network design model with a hub-and-spoke structure, where the market share is determined by cost and transit time. It is stressed that it is important to consider both transit time and cost in the construction of a liner shipping network.

### 1.4.3 Maritime Speed Optimization

A key decision related to the operation of a shipping network is the sailing speed between the serviced ports. As fuel costs may constitute more than 75% of the total operating cost of a vessel (Ronen, 2011) the sailing speeds across the



network has a significant impact on the operating costs of a network. Hence, it is also one of the planning problems where level of service considerations will impact the configuration substantially. Higher sailing speeds will yield shorter transit times for the customers but will at the same time result in a more expensive network operation illustrating the inherent trade-off in operating a low cost network versus a network which is optimized in terms of cargo transit times. Overall changes in sailing speed will also affect the strategic decisions regarding the required fleet size in the longer perspective.

Bunker consumption for a vessel profile is often modeled as a cubic function of speed (more on this in Chapter 5), but in practice it depends not only on the speed of operation, but also on the vessel type, the draft of the vessel (e.g., the actual load), the number of reefer containers powered by the vessel's engine, and weather conditions. During a round trip the vessel may sail at different speeds between ports. The vessel may *slow steam* to save bunker fuel or increase speed to meet a crucial transit time. Speed may be constrained not only by the vessel design, but also by hard weather conditions or navigation through difficult areas.

In the literature, Notteboom and Vernimmen (2009) provide detailed background information on the influence of bunker cost on liner shipping network configurations and describe how carriers changed sailing speed and number of vessels deployed when increasing bunker fuel cost increased. Ronen (2011) and Corbett et al. (2009) also conclude that the optimal sailing speed is likely to be reduced when the fuel price is increased. In Chapter 6 we indeed find this pattern. In addition to this Ronen (2011) present a model to determine the best average sailing speed and number of ships for a specific sequence of port calls. Wang and Meng (2012) consider a liner shipping network and present a non-linear mixed-integer programming model to optimize the speed of the deployed ships. However, container routing is only considered on a set of predefined container routes where all demand must be met, and no level of service requirements are considered. Kim (2013) present a model for a liner shipping service to minimize bunker costs and inventory costs for a single vessel. The model determines the bunkering ports as well as speed and number of vessels deployed on the service. No level of service requirements are imposed leading to very slow sailing speeds. There is a lack of methods for optimizing sailing speed on individual sailing legs in a network while considering level of service requirements and maximization profit. This is also the case for methods considering sailing optimization as part of a fleet deployment model. Gelareh and Meng (2010) develop a model to allocate ships to a set of predefined routes such that they minimize the fleet operating costs but they do not consider level of service requirements. They determine the frequencies and the sailing speeds

necessary to meet the demand. Meng and Wang (2011) also allocate ships and find sailing speeds for each leg in a single service while minimizing the daily operating costs, but again, no level of service requirements are imposed.

In the context of tramp shipping Wen et al. (2015) solve the simultaneous optimization of routing and sailing speed using a branch-and-price algorithm with heuristic column generation. The real-life test instances are all solved to (near) optimality in a short running time. Fagerholt et al. (2010) optimize the speed on each leg of a single fixed route where the sequence of ports and the time windows for each visited port are fixed. Norstad et al. (2011) and Gatica and Miranda (2011) solve the simultaneous speed optimization and routing problem for the less-than-shipload and full-shipload problems in tramp shipping. The routing and scheduling problem in tramp shipping is similar to the vehicle routing problem (Laporte et al., 2013). The same problem in liner shipping has very different characteristics, but some of the ideas can be adapted.

## 1.5 Contributions

This thesis contributes to the OR literature on liner shipping network optimization and continues the work within this field at DTU Management Engineering. The contributions of this thesis, which was written as part of the *Competitive Liner Shipping Network Design* project, are covering modeling, methodology, and application. It builds upon the work that started with the ENERPLAN (Energy Efficient Transportation Planning) project and especially the work by Berit Dangaard Brouer (Brouer, 2012), Christian Edinger Munk Plum (Plum, 2013), and Line Blander Reinhardt (Reinhardt, 2011). Hopefully, future contributions, including those through the GREENSHIP (Green Liner Shipping) project, will build upon the work presented in this thesis. In the following, we give an overview of the remaining chapters of this thesis, discuss the contributions, and give an overview of the dissemination of the work.

### 1.5.1 Maritime Optimization

**Part I, Introduction**, of this thesis motivates the problem at hand and presents a broader introduction to the use of quantitative decision support tools within maritime logistics in general and liner shipping in particular.

**Chapter 2, Big Data Optimization in Maritime Logistics**, gives an overview of some of the most important large-scale optimization problems faced by global carriers operating a network of container vessels. We show how decision support tools based on mathematical optimization techniques can guide the process of adapting a network to the current market and enable companies to get an overview of the decision process they have not previously had. The chapter is a review of previous work where the customer perspective is of less focus, but some of the ideas introduced here are the basis for the work done in this thesis. The content has been disseminated as follows:

- A book chapter co-authored with Berit Dangaard Brouer and David Pisinger, accepted in a forthcoming Springer book titled “Big Data Optimization” (Brouer et al., 2015b)
- Poster presentation by Christian Vad Karsten at *2014 MIT Transportation Showcase: “Transportation for Tomorrow”* at the MIT Museum
- Poster presentation by Christian Vad Karsten at *Maritime Researchers Day 2013: “New Maritime Initiatives”* at Copenhagen Business School

### 1.5.2 Competitive Liner Shipping Network Design

**Part II, Network Optimization with Transit Time Restrictions**, of this thesis explores to best include travel time restrictions on the cargo routing in network optimization algorithms and further analyze the impact on the obtained solutions and network configurations.

The time constrained multi-commodity network flow problem is formulated and tailored for the liner shipping application in **Chapter 3, The Time Constrained Multi-commodity Flow Problem**. We present a tailored column generation algorithm for the liner shipping application and show that it is possible to include transit time restrictions without hurting the computational performance. The novelty of the column generation algorithm includes solving a resource constrained shortest path problem using a specialized label setting algorithm which only needs to be solved once for each origin. The computational results show that the algorithm scales well to larger problems. Therefore, it is valuable in itself for planners in order to evaluate the impact of proposed changes in a network, but can also be used as part of network design algorithms as shown in the following chapters. As discussed previously it is generally acknowledged that transit times are decisive for the competitiveness of a network. This leads to a network design process that allows for multiple objectives as the customers must balance minimal transit times against low freight rates. The results in this chapter indeed show that providing low freight rates by minimizing the cost of the network is likely to result in prolonged transit times. Finally, different graph topologies tailored for liner shipping networks that can handle different levels of details related to port operations are presented. These topologies make it possible to extend the discussed algorithms to handle a time schedule and include port productivity considerations. The work has been disseminated as follows:

- A paper co-authored with David Pisinger, Stefan Røpke, and Berit Dangaard Brouer, published in *Transportation Research Part E: Logistics and Transportation Review* (Karsten et al., 2015f)
- Presentation by Christian Vad Karsten at *26th European Conference on Operational Research* (Karsten et al., 2013)

We utilize the solution method of the time constrained cargo routing problem presented in Chapter 3 to build a network design algorithm in **Chapter 4, Time Constrained Liner Shipping Network Design**. The presented algorithm extends previous work by Brouer et al. (2014b) and is the first step

to take level of service considerations into account in the design process. The core of the algorithm is an improvement heuristic, where an integer program is solved iteratively as a move operator in a large neighborhood search. To assess the effects of insertions/removals of port calls, cargo flow, and revenue changes are estimated for relevant commodities along with an estimation of the change in the vessel cost. The estimation functions leverage that the column generation technique described in the previous chapter can be warm started using previously generated columns leading to very effective updates when assessing flow changes. The results are promising as highly profitable networks can be generated, but they also indicate that the constant speed usually considered for the routes in the network design process needs to be extended to variable speed on all sailing legs to properly leverage the potential of network design algorithms. In the chapter we additionally show that it is possible to introduce limits on the number of transshipments in the proposed algorithm. The transit time and transshipment constrained algorithm is able to produce networks that are as profitable as the networks where only transit time is restricted. The work has been disseminated as follows:

- A conference paper co-authored with Berit Dangaard Brouer, Guy Desaulniers, and David Pisinger has been published in *Computational Logistics* (Brouer et al., 2015a). The conference paper was selected among the best contributions and invited for publication in an extended version.
- A paper co-authored with Berit Dangaard Brouer, Guy Desaulniers, and David Pisinger, under review in special issue of *Transportation Research Part E* on “Coordination and Control in Transport Logistics” (Karsten et al., 2015c)

To address the need for speed optimization we develop a method for simultaneous optimization of vessel speed and container routing with transit time restrictions in **Chapter 5, Simultaneous Optimization of Sailing Speed and Container Routing with Transit Time Restrictions**. The model is formulated over an augmented network to handle the cargo transit time restrictions. This makes it possible to optimize the entire network or specific regions. The formulation of the model makes the incorporation of an actual schedule straightforward to reflect transshipment times more accurately. Furthermore, we present an extension of the model that makes it possible to take an actual schedule into account and optimize the entire schedule or part of it to get the best coordination between services. The solution method relies on a column and row generation (Benders decomposition) procedure that leverage the separability of the problem. Again, we exploit that we can warm start when re-solving the flow problem to speed up the algorithm. Further algorithm-

mic enhancements are based on valid inequalities and by utilizing the callback capabilities of modern MIP solvers. The results clearly show the improvement potential, but the developed method is more appropriate for post-processing of networks than being part of an actual network design algorithm due to the computational effort needed. The results show that variable speed on each sailing leg and corresponding changes in fleet deployment can lead to large savings in operational costs and profit improvements of more than 10 % are found compared to a network operated at constant speed. One important finding is that it indeed is important to consider the cargo flow as part of the speed optimization process. We show that speed changes can lead to rather significant changes in the routing of the containers, and hence we conclude that it is important to consider routing implications when optimizing speed in networks where transit time restrictions on cargo paths are imposed. Previous models frequently follow a strategy where the routes for the containers are determined in an initial stage. Then, in a second stage, the actual planning of sailing speed and schedules takes place using the knowledge of which routes containers already use. However, the actual routes that will be attractive for a shipper will strongly depend on the offered transit time. Hence, it seems to be necessary to consider both stages in a more holistic approach. The work has been disseminated as follows:

- A paper co-authored with Stefan Røpke and David Pisinger, under review in *Transportation Science* (Karsten et al., 2015d). Also published as a technical report from DTU (Karsten et al., 2015e).

Following these results we develop a very efficient method to include speed optimization in the network design process in **Chapter 6, Competitive Liner Shipping Network Design**, while still considering transit time restrictions and cargo routing. This method is incorporated in the algorithm presented in Chapter 4 and, with some additional adjustments to better account for the most attractive moves and speed changes during the design process, we present a solution method for the competitive liner shipping network design problem in Chapter 6. The proposed algorithm is intended for incremental optimization of existing networks. Furthermore, we show that it is capable of building a new competitive network from scratch by improving an initial set of services generated by a simple heuristic that selects a set of ports, with extensive mutual trade, forming a good service. The computational results are encouraging and for all larger instances considered, the networks generated with variable speed are highly profitable and consistently better than a network operated at constant speed. The average profit improvement is up to 10 % with a maximum improvement of 60 % in the best case. Finally, a bunker sensitivity study shows the expected trend that speed appear to decrease with increased bunker price

(0.2 nm/h per 100 \$/ton increase). Correspondingly, deployment increase with increased bunker price (0.2 % per 100 \$/ton increase). The amount of transported demands decrease with increased bunker price (0.8 % per 100 \$/ton increase) showing that it may not be profitable to meet transit times for all demands even with different network layouts.

The algorithm for the time and transshipment constrained multi-commodity flow problem can easily be incorporated in the solution method for the competitive liner shipping network design problem discussed in Chapter 6 to design networks taking both transit time and number of transshipments into account described in Chapter 4. However, on the input side the number of transshipments allowed for each cargo will influence the performance, but with tight limits, the performance of the algorithm can be expected to improve.

The work has been disseminated as follows:

- A paper co-authored with Berit Dangaard Brouer and David Pisinger is under review in *Computers and Operations Research* (Karsten et al., 2015a). Also published as a technical report from DTU (Karsten et al., 2015b).

### 1.5.3 Service Selection with Limited Transshipments

**Part III, Service Selection with Limited Transshipments**, takes an alternative view on network design and the introduction of level of service requirements. One way to introduce level of service considerations in the network design is by limiting the travel time for the cargo. As discussed, an alternative approach is to limit the number of transshipments. In this part of the thesis we present two models that incorporate limits on the number of transshipments to reflect common practice in liner shipping networks. These limits are analogous to the hop constraints introduced by Balakrishnan and Altinkemer (1992). The presented planning models have the ability to incorporate demand-specific limits on the number of transshipments. Furthermore, rather than designing services we formulate a problem similar to line planning problems discussed for the public transportation setting (Schöbel, 2012). In both of the presented models it is straightforward to include operational policies such as cabotage rules and embargoes. In certain areas of the world such rules have a large impact on the network design and should be considered in the design process as it may be limiting in terms of feasible flows. At the same time the models are formulated such that the inclusion of transshipment limits and operational policies helps to reduce the size of the models. We model the problem as one of selecting services from a candidate set of services provided by either an ex-

perienced planner or an algorithm. This approach is appropriate for tactical planning whenever demand patterns, operating costs, or economic conditions change. In these situations, the planner often prefers to limit the service choices and changes based on the current network. Moreover, the option to specify candidate services permits planners to incorporate not only economic factors but also strategic, operational, political, competitive, and other issues that may not be easy to represent in optimization models.

**Chapter 7, Container Shipping Service Selection and Cargo Routing with Limited Transshipments**, presents a model addressing the problem of selecting which services to operate from a potentially large pool of candidate services so as to maximize profit. The corresponding solution method is simple to implement and it takes advantage of the capabilities of contemporary integer programming solvers for solving large-scale problem instances. The service selection is done while considering cargo routing with limited transshipments. We propose a hop-constrained multi-commodity arc flow model that is based on an augmented network containing, for each candidate route, an arc (representing a sub-path) between every pair of ports visited on the route. This sub-path construction permits us to accurately model transshipment costs and incorporate routing policies. The work has been disseminated as follows:

- A paper co-authored with Anantaram Balakrishnan, under review in a special issue of *Annals of Operations Research* on Logistics, Optimization, and Transportation (Balakrishnan and Karsten, 2015a)
- Presentation by Christian Vad Karsten at *LOT - Logistics, Optimization and Transportation : A special EU/MEeting in memory of late Professor Arne Løkketangen* (Karsten and Balakrishnan, 2014a)
- Presentation by Christian Vad Karsten at *INFORMS Annual Meeting 2014* (Karsten and Balakrishnan, 2014b)

In **Chapter 8, Optimal Selection of Liner Containership Services with Limited Transshipments**, we improve the model presented in Chapter 7 by formulating it over a multi-layer network where we can still exploit the capabilities of integer programming solvers for solving large-scale problem instances. Instead of defining a separate commodity for each demand, we define one commodity for all the containers that originate at a port node. We then “decompose” this commodity into flows for individual O-D demands by tracking the commodity’s outflow from the system at the relevant destinations. We propose a model that uses a representation of the container movements and transfers as multi-commodity flows over a logical network that permits capturing the



transshipment costs and restrictions. We then assign these flows to the chosen liner shipping services, in the physical network, taking into account ship capacities and operating costs. The logical layer is based on segments. A segment is defined between two ports only if there is at least one service that visits both ports. A segment is a link in the “logical” network layer and for each segment, the physical layer may contain several services that can carry the traffic on this segment. This modeling approach reduces the number of variables and constraints relative to the previous model without sacrificing the tightness of the linear programming relaxation. We are able to solve larger instances using this approach. The work has been disseminated as follows:

- A paper co-authored with Anantaram Balakrishnan, submitted to *European Journal of Operations Research*.
- Presentation by Christian Vad Karsten at *27th European Conference on Operational Research* (Karsten and Balakrishnan, 2015)

Generally, size and strength of a formulation are the main determinants for the performance of state of the art branch-and-bound solvers. The two formulations presented in Chapter 7 and 8 model the same problem and have the same LP-relaxation in the base version. However, the disaggregated model presented in Chapter 7 is significantly larger than the aggregated model presented in Chapter 8. It is straightforward to strengthen the model in Chapter 7 by the use of forcing constraints, while these are less effective for the aggregated version of the model. Still, the significantly smaller size of the model discussed in Chapter 8 offset the strength of the larger model in the computational results provided.

The work presented in this thesis presents profit maximization models (rather than cost minimization which is often considered in the literature) and corresponding solution methods for operational, tactical, and strategic planning problems faced by liner shipping companies. The models consider both the carrier and shipper perspective, leading to advanced tools that can support the decision process related to operating a competitive liner shipping network.

## 1.6 Conclusions

The work in this thesis contributes to the OR literature on liner shipping network optimization and has been disseminated in peer-reviewed journals and conferences. The contributions cover modeling, methodology, and applications. They are concerned with integrating level of service requirements in large-scale mathematical models addressing operational (cargo routing), tactical (speed optimization and service selection), and strategic (network design) planning problems faced by liner shipping companies. The level of service offered to the shipper is determined by several factors but the most important aspects we consider are the combination of cost, transit time, and number of transshipments.

The global container transportation network has grown to a size where humans alone cannot easily assess the best configuration and hence they must rely on advanced analytics in the decision process to attain the most competitive network. Using the advanced planning tools presented in this thesis, a liner shipping company will quickly be able to assess the impact of their decisions and get an overview of the ripple effects across the network. The presented network design algorithm can suggest the best improvements to one or several sailing routes while still considering the effects across the entire network. From a practical perspective a tool to support incremental changes and improvements is of high value as it can help managers and planners to quickly adapt to new situations as well as when negotiating cooperation and VSA. The algorithms are designed to maximize profit from the point of view of the carrier while respecting the view of the shipper in terms of level of service requirements in order to attain a balance between the two dual perspectives. The models developed for the tactical and strategic planning problems of speed optimization, service selection, and network design, are all integrated with the solution of the operational cargo routing problem as this has been shown to be extremely important to be able to design a competitive network.

The contributions of this thesis has achieved a level of detail for the network design algorithm presented in Part II such that it is reaching a state where it is applicable as a decision support tool in a real world setting. The most important shipper perspectives are considered. Currently, we are collaborating with Maersk Line on tailoring the algorithm to their network such that it can gradually start to contribute actively to the decision process. The analysis tools developed in Chapter 6 can help further interpret the performance when tested in a real world setting. The cargo flowing sub-problem that needs to be solved for a global network is larger and less aggregated than the multi-commodity flow problems solved in Chapter 3. However, for the real network

we see acceptable running times so far when solving the cargo routing problem, making it realistic to also apply the network design algorithm. Based on the current network operated by Maersk Line (or any other carrier) the decision support tool can eventually help understand how the capacity utilization of the network can be improved while considering customer requirements. The tool can provide insights on the design of new routes and make suggestions for the planners such that they utilize existing and new markets, and possibly leave unprofitable markets. Furthermore, it will be possible for the planners as well as strategic decision makers to create different scenarios and use the tool to better understand how changes in the fleet can best be integrated into the existing network and what ships will be relevant to use in the future.

## 1.7 Future Work

In the following we outline some possible extensions for the models in this thesis to be more likely to be adapted by the industry and possible directions for future research related to modeling, applications and methodology.

### 1.7.1 Improved Modeling

The methods described in this thesis can be extended to handle additional details such that the flow solutions obtained from the cargo routing problem to a higher degree reflect actual operations. Here an additional level of detail may be needed in the flow evaluations. In the multi-commodity flow problem used in Chapters 3 - 6 the restrictions discussed in the sequel can be imposed relatively easily by simple graph modifications and are mainly concerned with operational alliances and legislation.

As discussed, VSA are central to liner shipping operations and with increasing size of the vessels deployed they become central to incorporate as part of the decision process. A VSA is straightforward to model in the flow problems by simply using a reduced capacity on certain sailing legs compared to the deployed vessel class. In the network design algorithms certain edges can be fixed such that they are not available to change. An alternative approach will, through more careful analysis of the dual information obtained from these edges, potentially help to put a value on these. Eventually, this may be very valuable during negotiations with partners and internally give an indication of which strategies to pursue. Similarly, modeling FFE only requires additional edges in routing problem with the FFE related cost and fixed edges in the network design problem. Like the VSA it may be possible to better assess the value of a given FFE. Legislative aspects may also be included in the flow evaluations by a slight modification of the underlying graph. Sulphur emission control areas, SECA, or emission control areas, ECA, are areas in the sea where stricter controls are established to minimize airborne emissions such as SO<sub>x</sub> (sulphur oxides), NO<sub>x</sub> (nitrogen oxides), ODS (ozone depleting substances), and VOC (volatile organic compounds) from ships. The impact of these on the network design varies from region to region, but one way to handle them in the design process is to have two alternative available routes/edges in the graph (as in Fagerholt and Psaraftis (2015)) in relevant areas. Edges through and around the ECA with the corresponding cost of the fuel will allow the algorithms to decide the best option. The canals are already handled this way, e.g. for the Suez there is an alternative much longer sailing edge avoiding the canal. Along the same lines, cabotage rules apply to commercial ships in most

countries and are implemented to protect the domestic shipping industry from foreign competition, hence these are quite important to consider both during the design process and especially when evaluating flows. Cabotage rules are further discussed in Part III of this thesis and here they are already handled as part of the pre-processing so all container routings automatically satisfy these. In the path flow formulation of the multi-commodity flow problem discussed in Part II, they can be implemented by including an additional resource which is incremented in “illegal” ports. Finally, a more detailed cargo mix (reefers, high cube, 20’, 40’, etc.) can also be handled in the cargo routing problem by including additional capacity constraints in the multi-commodity flow problem and a specification of the container type in the demand list. Hence, it increases the size of the problem, but the size should still be manageable.

Level of Service requirements can be formulated in terms of a maximum transit time or limiting the number of transshipments as done in this thesis. Alternatively time windows (as used in tramp shipping) or inventory costs can be considered as proxies for level of service (Fagerholt et al., 2010; Norstad et al., 2011; Hvattum et al., 2013; Álvarez, 2012; Kim, 2013). Likewise, the maximum transit time can be modeled as soft constraints which could be punished for exceeding rather than hard constraints, or the overall optimization could be done in a multi-objective setting where both cost, transit time, and any other level of service requirements are optimized.

In Part II only transit times are considered. However, as shown in the sensitivity study in Chapter 4 the path flow formulation of the multi-commodity flow problem can easily be extended to also explicitly limit the number of transshipments by including a resource that keeps track of the number of transshipment edges used. In Part III it is harder to extend the models to explicitly handle transit time in addition to limiting the number of transshipments. Additional variables that explicitly keep track of the time can be included but this would likely increase the size of the models to make them intractable. However, in the model discussed in Chapter 7, the flow variables are associated with a specific service. Therefore, it is possible, as part of the pre-processing, to guarantee that a demand is only considered if there is a least one path satisfying the maximum allowed transit time for this demand. Hence, all sub-paths (or combinations of these) that explicitly violates transit time restrictions can be removed from the model. However, this does not guarantee that the actual path used does not violate transit time.

Both from a computational perspective and level of detail considered it is still relevant to consider whether to pursue arc- or path flow formulations. Table 1.3 shows computational results for different levels of service and algorithms used for solving the cargo routing problem. Here we have extended the cargo routing

Instance	Algorithm	L.S.	Transp.	CPU time
$WS_1$	col. gen. (path)	none	91.5 %	9.4 s
$WS_2$	col. gen. (path)	none	89.6 %	11.4 s
$WS_1$	col. gen. (path)	tt.	87.6 %	3.3 s
$WS_2$	col. gen. (path)	tt.	86.9 %	3.8 s
$WS_1$	LP (arc)	ts.	91.0 %	7.4 s
$WS_2$	LP (arc)	ts.	89.4 %	6.7 s
$WS_1$	col. gen. (path)	tt.+ts.	87.5 %	2.7 s
$WS_2$	col. gen. (path)	tt.+ts.	86.8 %	2.8 s

**Table 1.3:** Runtimes of solving the cargo routing problem for different models, algorithms, and different level of service, L.S. A restriction on transit time is denoted *tt.* and a limit on the number of transshipments is *ts.* Networks are generated using the algorithm presented in Chapter 4 using Liner-lib data (Brouer et al., 2014a) for WorldSmall, WS, instances. Transit time restrictions are given in the data set. When restricting the number of transshipments the limit is set to two for all cargoes. The *complete* transshipment structure described in Chapter 3 is used in the path flow formulation.

problem in the path flow formulation to include restrictions on the number of transshipments as done in Chapter 4. Solving the sub-problem with tight limits on both transit time and number of transshipments is 15-20 % faster than when only considering transit time. Though the shortest path algorithm is extended with an additional resource, the labels are limited sufficiently to make the algorithm terminate faster. This is an indication that the level of detail considered in the cargo flow can help restrict the solution space sufficiently to offset the increased complexity in the resource constrained shortest path sub-problem and turns out to improve dominance significantly. Compared to the path flow formulation with no level of service requirements imposed it is seen that the effect is quite substantial. However, it is not clear whether this trend will continue with incorporation of additional resources modeling e.g. cabotage rules in the path flow formulation. In the presented arc-flow formulation cabotage rules in fact helps reducing the size of the models and generally we have shown how policy based rules can help manage model sizes, but in the current setup the path flow formulations seems to be superior. However, it is worth noticing that the networks considered in Table 1.3 are designed under consideration of transit time restrictions using the algorithm described in Chapter 4 while the networks considered in Chapter 3 are designed using a version of the algorithm without considering transit times. A more careful comparison of these shows that this also influence computational times.

The more detailed the level of service aspects can be considered in the decision

support tools the more tailored products can be offered to customers and it is possible to differentiate customers according to level of service. Consequently it will be possible for a liner shipping company to request higher transportation costs from customers and possibly even attract new customers.

### 1.7.2 Additional Applications

As discussed in Section 1.4.1 the cargo routing problem is at the heart of many of the planning problems in liner shipping and it needs to be evaluated often. We have proposed a solution method that effectively solves the problem while considering transit time restrictions and this can be the basis for integrating level of service requirements in many of the planning problems encountered. The cargo routing problem is also closely related to the problem of *repositioning empty containers* through the network. In empty repositioning the cost and number of transshipments are obviously more important than transit times as the shipper usually do not care about empty containers. The problem can be addressed by including additional demands in the proposed column generation algorithm, but from a practical perspective it may be more attractive to be able to implement tools using off-the-shelf solvers by considering the flow part of the models discussed in Chapters 7 and 8 with the integer variables corresponding to service selection fixed. This gives an LP model that directly minimizes cost and restricts the number of transshipments (which in this case is more important than travel times). The model does not require more advanced implementations such as Benders decomposition.

The network design algorithms in their current form has applications much broader than pure strategic network design. As discussed previously, carriers need tools to decide whether to deliver containers using direct connections or transship at an interim hub and whether to own or lease the fleet. In addition to these decisions, bunkering is a very important aspect of the operating to consider to get the most profitable network. As shown by Plum et al. (2015), there are significant variations in bunker prices across the world. This means that on a given service it may be attractive to call a port with cheap bunkering options. This adds a new dimension to the network design process. As the proposed algorithms already aim at incremental network design this would be a natural extension, where they could be used in a tactical or even operational setting to consider bunkering options in the current network design.

### 1.7.3 Extensions of Methods

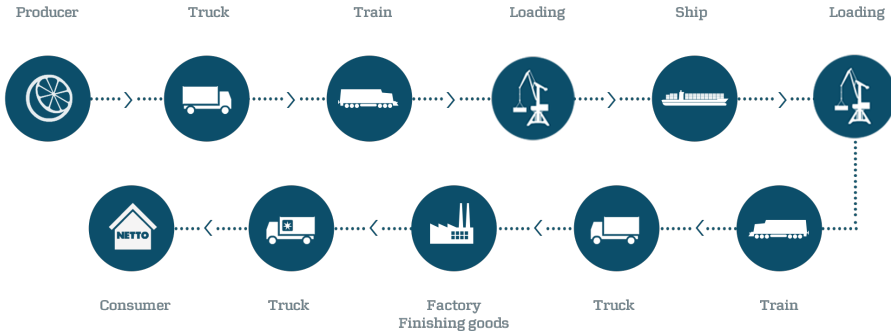
There are several ways to extend or even improve the methods presented in Part III of this thesis. From a planner perspective this could yield faster methods making the decision process more agile. From a shipper perspective better solutions will make the carriers able to offer more competitive products at the same or even lower price. Related to the cargo routing problem it is relevant to consider methods that can help speed up the solution as the problem is evaluated often. The “tailing off effect” is limited when solving the cargo flow-ing multi-commodity flow problem discussed in Chapter 3 but a stabilization scheme on the dual variables may still improve the convergence (Lübbecke and Desrosiers, 2005). Also, it would be interesting to test alternative solution methods for the multi-commodity flow problem. Babonneau et al. (2006) propose a method with good performance when bottlenecks appear in a network, which is the case for several sailing legs in a global network. The method is based on a Lagrangian relaxation restricted to the arcs that are likely to be saturated at the optimum. These will be relatively easy to identify in a liner shipping network, e.g. canals and head haul on main trades and could be competitive alternative to the proposed method in terms of computational time, but it may be more complicated to consider level of service requirements.

The search space used in the matheuristics for liner shipping network design described in Chapter 4 and 6 is rather constrained and guided by feasible solutions. A possible improvement to the heuristic in the domain where new networks are designed from scratch would be to allow the initial phase of the search to explore infeasible areas of the search space. By starting out with a less constrained fleet in terms of availability and zero cost of deploying vessels then the algorithm would have more freedom in terms of designing routes. Then by slowly letting the cost converge towards the real cost and the fleet size converge towards the real fleet, services may be constructed that have a better capacity utilization.

One aspect of the search space currently not explored in the matheuristic presented in Chapter 6 is related to capacity and vessel classes. An additional neighborhood to consider would be to allow to swap or replace the vessel class on a service with another.

The solution method for the model discussed for speed optimization in Chapter 5 is based on Benders decomposition. This solution method still shows potential for improving the convergence. We have already tested ideas along the lines of Fischetti et al. (2015) where we slightly perturb the point to cut off before invoking the separator. However, we did not see any significant effects, but the strategy for perturbing can still be further explored along these lines as Fischetti





**Figure 1.10:** Global supply chain. *Source: Maersk (2015).*

et al. (2015) show very impressive results. From a practical perspective, it may be of more interest to find good solutions quickly rather than having guarantees on the quality of these. Improvements in this direction could be along the lines of integrating ideas from recent work on proximity search (Fischetti and Monaci, 2014). Proximity search complements Benders decomposition as the aim of this heuristic is to iteratively produce a sequence of improving solutions by solving a slightly modified problem (Bolanda et al., 2015).

The models presented in Part III for service selection has the advantage that they can be solved using off-the-shelf MIP solvers and therefore the adoption of these is easy for companies. However, the structure of the problems is such that they could be solved using e.g. Benders decomposition, which potentially could improve computational performance. Additionally, it would be interesting to compare the proposed heuristics which are based on either LP-relaxations or reduced problems based on problem characteristics with a proximity based search approach (Fischetti and Monaci, 2014) as this also takes advantage of standard solver capabilities.

Finally, it is important to remember that the shipping part is only a part of the global supply chain as illustrated by Figure 1.10 and all links can introduce delays and uncertainty. Therefore, decision support tools should, both from a carrier and customer perspective, ideally integrate uncertainty and robustness considerations and manage a larger part of the supply chain to be able to eventually offer better and more robust products to both producers and consumers. This will require additional extensions of the application, modeling, and methodology presented in this thesis.

## Bibliography

- Agarwal, R. and Ergun, Ö. (2008a). Mechanism design for a multicommodity flow game in service network alliances. *Operations Research Letters*, 36(5):520–524.
- Agarwal, R. and Ergun, Ö. (2008b). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196.
- Agarwal, R. and Ergun, Ö. (2010). Network design and allocation mechanisms for carrier alliances in liner shipping. *Operations Research*, 58(6):1726–1742.
- Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.
- Alderton, P. M. (2011). *Reeds sea transport: operation and economics*. A&C Black.
- Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11(2):186–208.
- Álvarez, J. F. (2012). Mathematical expressions for the transit time of merchandise through a liner shipping network. *Journal of the Operational Research Society*, 63(6):709–714.
- Andersson, H., Hoff, A., Christiansen, M., Hasle, G., and Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536.
- Babonneau, F., Du Merle, O., and Vial, J.-P. (2006). Solving large-scale linear multicommodity flow problems with an active set strategy and proximal-accpm. *Operations Research*, 54(1):184–197.
- Baird, A. J. (2006). Optimising the container transshipment hub location in northern europe. *Journal of transport geography*, 14(3):195–214.
- Balakrishnan, A. and Altinkemer, K. (1992). Using a hop-constrained model to generate alternative communication network design. *ORSA Journal on Computing*, 4(2):192–205.
- Balakrishnan, A. and Karsten, C. V. (2015a). Joint service selection and cargo routing with limited transshipments. Submitted to: *Annals of Operations Research: Logistics, Optimization, and Transportation*.
- Balakrishnan, A. and Karsten, C. V. (2015b). Optimal selection of liner containership services with limited transshipments. Submitted to: *European Journal of Operations Research*.

- Balakrishnan, A., Magnanti, T. L., and Mirchandani, P. (1997). Network design. *Annotated bibliographies in combinatorial optimization*, pages 311–334.
- Barnhart, C., Hane, C. A., and Vance, P. H. (2000). Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326.
- BBC (2015a). Are we seeing the end of the era for the 'super-jumbo'? <http://www.bbc.com/news/business-32384435>.
- BBC (2015b). The invisible network that keeps the world running. <http://www.bbc.com/future/story/20150209-the-network-that-runs-the-world>.
- Bierwirth, C. and Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615–627.
- Bolanda, N., Fischettib, M., Monacib, M., and Savelsbergha, M. (2015). Proximity benders: A decomposition heuristic for stochastic programs. Technical report, University of Padova.
- Borndörfer, R., Grötschel, M., and Pfetsch, M. E. (2007). A column-generation approach to line planning in public transport. *Transportation Science*, 41(1):123–132.
- Borndörfer, R. and Karbstein, M. (2012). A direct connection approach to integrated line planning and passenger routing. In *ATMOS*, pages 47–57.
- Brouer, B. D. (2012). *Liner Service Network Design*. PhD thesis, Technical University of Denmark.
- Brouer, B. D., Álvarez, J. F., Plum, C. E., Pisinger, D., and Sigurd, M. M. (2014a). A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312.
- Brouer, B. D., Desaulniers, G., Karsten, C. V., and Pisinger, D. (2015a). A matheuristic for the liner shipping network design problem with transit time restrictions. In *Computational Logistics*, pages 195–208. Springer.
- Brouer, B. D., Desaulniers, G., and Pisinger, D. (2014b). A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59.
- Brouer, B. D., Karsten, C. V., and Pisinger, D. (2015b). Big data optimization in maritime logistics. In Emrouznejad, A., editor, *Big Data Optimization*, page Forthcoming. Springer.

- Brouer, B. D., Pisinger, D., and Spoorendonk, S. (2011). Liner shipping cargo allocation with repositioning of empty containers. *INFOR: Information Systems and Operational Research*, 49(2):109–124.
- Brynjolfsson, E., Hitt, L. M., and Kim, H. H. (2011). Strength in numbers: How does data-driven decisionmaking affect firm performance? Available at SSRN 1819486.
- Caprara, A., Kroon, L., Monaci, M., Peeters, M., and Toth, P. (2007). Passenger railway optimization. *Handbooks in operations research and management science*, 14:129–187.
- Christiansen, M. and Fagerholt, K. (2009). Maritime inventory routing problems maritime inventory routing problems. In *Encyclopedia of optimization*, pages 1947–1955. Springer.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2007). Maritime transportation. *Handbooks in operations research and management science*, 14:189–284.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Christiansen, M., Fagerholt, K., and Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation science*, 38(1):1–18.
- Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2013). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.
- Corbett, J. J., Wang, H., and Winebrake, J. J. (2009). The effectiveness and costs of speed reductions on emissions from international shipping. *Transportation Research Part D: Transport and Environment*, 14(8):593–598.
- Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288.
- Crainic, T. G., Kim, K. H., et al. (2006). Intermodal transportation. *Transportation*, 14:467–537.
- Dantzig, G. and Fulkerson, D. (1954). Minimizing the number of tankers to meet a fixed schedule. *Naval Res. Logist. Quart.*, 1:217–222.
- Davenport, T. (2014). *Big data at work: dispelling the myths, uncovering the opportunities*. Harvard Business Review Press.
- Davenport, T. H. and Harris, J. G. (2007). *Competing on analytics: The new science of winning*. Harvard Business School Press.

- De Botton, A. (2010). *The pleasures and sorrows of work*. Emblem Editions.
- Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. (1995). Time constrained routing and scheduling. *Handbooks in operations research and management science*, 8:35–139.
- Desrosiers, J. and Lübbecke, M. E. (2005). *A primer in column generation*. Springer.
- Fagerholt, K. (1999). Optimal fleet design in a ship routing problem. *International Transactions in Operational Research*, 6(5):453–464.
- Fagerholt, K., Laporte, G., and Norstad, I. (2010). Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3):523–529.
- Fagerholt, K. and Psaraftis, H. N. (2015). On two speed optimization problems for ships that sail in and out of emission control areas. *Transportation Research Part D: Transport and Environment*, 39:56–64.
- Fischetti, M., Ljubic, I., and Sinnl, M. (2015). Thinning out facilities: a benders decomposition approach for the uncapacitated facility location problem with separable convex costs. Technical report, University of Padova.
- Fischetti, M. and Monaci, M. (2014). Proximity search for 0-1 mixed-integer convex programming. *Journal of Heuristics*, 20(6):709–731.
- Flood, M. M. (1954). Application of transportation theory to scheduling a military tanker fleet. *Journal of the Operations Research Society of America*, 2(2):150–162.
- Gatica, R. A. and Miranda, P. A. (2011). Special issue on latin-american research: A time based discretization approach for ship routing and scheduling with variable speed. *Networks and Spatial Economics*, 11(3):465–485.
- Gelareh, S. and Meng, Q. (2010). A novel modeling approach for the fleet deployment problem within a short-term planning horizon. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):76–89.
- Gelareh, S., Nickel, S., and Pisinger, D. (2010). Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):991–1004.
- Gendron, B., Crainic, T. G., and Frangioni, A. (1999). *Multicommodity capacitated network design*. Springer.
- Glave, T., Joerss, M., and Saxon, S. (2014). The hidden opportunity in container shipping. Technical report, McKinsey & Company.

- Grønhaug, R. and Christiansen, M. (2009). Supply chain optimization for the liquefied natural gas business. *Innovations in distribution logistics*, 619:195–218.
- Halpern, B. S., Walbridge, S., Selkoe, K. A., Kappel, C. V., Micheli, F., D’Agrosa, C., Bruno, J. F., Casey, K. S., Ebert, C., Fox, H. E., et al. (2008). A global map of human impact on marine ecosystems. *Science*, 319(5865):948–952.
- Hemmati, A., Hvattum, L. M., Fagerholt, K., and Norstad, I. (2014). Benchmark suite for industrial and tramp ship routing and scheduling problems. *INFOR: Information Systems and Operational Research*, 52(1):28–38.
- Hoff, A., Andersson, H., Christiansen, M., Hasle, G., and Løkketangen, A. (2010). Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061.
- Hvattum, L. M., Norstad, I., Fagerholt, K., and Laporte, G. (2013). Analysis of an exact algorithm for the vessel speed optimization problem. *Networks*, 62(2):132–135.
- Joerss, M., Murnane, J., Saxon, S., and Widdows, R. (2015). Landside operations: The next frontier for container-shipping alliances. Technical report, McKinsey & Company.
- Karsten, C. V. and Balakrishnan, A. (2014a). Modeling and solving the liner shipping service selection problem. In *LOT-Logistics, Optimization and Transportation*.
- Karsten, C. V. and Balakrishnan, A. (2014b). Planning liner shipping services. In *INFORMS Annual Meeting 2014*.
- Karsten, C. V. and Balakrishnan, A. (2015). Modeling liner shipping service selection and container flows using a multi-layer network. In *27th European Conference on Operational Research*.
- Karsten, C. V., Brouer, B. D., and Pisinger, D. (2015a). Competitive liner shipping network design. Submitted to: *Computers and Operations Research*.
- Karsten, C. V., Brouer, B. D., and Pisinger, D. (2015b). Competitive liner shipping network design. Technical report, Technical University of Denmark.
- Karsten, C. V., Pisinger, D., Brouer, B. D., and Desaulniers, G. (2015c). Time constrained liner shipping network design. Submitted to: *Transportation Research Part E: Coordination and Control in Transport Logistics*.

- Karsten, C. V., Pisinger, D., and Røpke, S. (2013). An algorithm for solving the time-constrained multicommodity flow problem with applications in liner shipping network design. In *26th European Conference on Operational Research*.
- Karsten, C. V., Pisinger, D., and Ropke, S. (2015d). Simultaneous optimization of liner shipping vessel speed and container routing with transit time restrictions. Submitted to: *Transportation Science*.
- Karsten, C. V., Pisinger, D., and Ropke, S. (2015e). Simultaneous optimization of liner shipping vessel speed and container routing with transit time restrictions. Technical report, Technical University of Denmark.
- Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015f). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138.
- Kim, H. (2013). A lagrangian heuristic for determining the speed and bunkering port of a ship. *Journal of the Operational Research Society*, 65(5):747–754.
- Kjeldsen, K. H. (2011). Classification of ship routing and scheduling problems in liner shipping. *INFOR: Information Systems and Operational Research*, 49(2):139–152.
- Krugman, P. (2013). Should slowing trade growth worry us? *New York Times*.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.
- Laporte, G., Toth, P., and Vigo, D. (2013). Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics*, 2(1-2):1–4.
- Levinson, M. (2010). *The box: how the shipping container made the world smaller and the world economy bigger*. Princeton University Press.
- Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.
- Maersk (2015). Transport & trade. <http://www.maersk.com/en/industries/transport>.
- Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation science*, 18(1):1–55.
- McAfee, A. and Brynjolfsson, E. (2012). Big data: The management revolution. *Harvard Business Review*, 90(10):61–67.

- Meng, Q. and Wang, S. (2011). Optimal operating strategy for a long-haul liner service route. *European Journal of Operational Research*, 215(1):105–114.
- Meng, Q., Wang, S., Andersson, H., and Thun, K. (2014). Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280.
- NCEAS (2008). A global map of human impacts to marine ecosystems. <https://www.nceas.ucsb.edu/globalmarine/impacts>.
- Norstad, I., Fagerholt, K., and Laporte, G. (2011). Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19(5):853–865.
- Notteboom, T. and Rodrigue, J.-P. (2008). Containerisation, box logistics and global supply chains: The integration of ports and liner shipping networks. *Maritime Economics & Logistics*, 10(1):152–174.
- Notteboom, T. E. and Vernimmen, B. (2009). The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337.
- OECD, T. (2015). The impact of mega-ships. Technical report, International Transport Forum.
- Papageorgiou, D. J., Nemhauser, G. L., Sokol, J., Cheon, M.-S., and Keha, A. B. (2014). Mirplib—a library of maritime inventory routing problem instances: Survey, core model, and benchmark results. *European Journal of Operational Research*, 235(2):350–366.
- Plum, C. E., Pisinger, D., and Jensen, P. N. (2015). Bunker purchasing in liner shipping. In *Handbook of Ocean Container Transport Logistics*, pages 251–278. Springer.
- Plum, C. E., Pisinger, D., and Sigurd, M. M. (2014). A service flow model for the liner shipping network design problem. *European Journal of Operational Research*, 235(2):378–386.
- Plum, C. E. M. (2013). *Optimization of Container Line Networks with Flexible Demands*. PhD thesis, Technical University of Denmark.
- Porter, M. E. and Heppelmann, J. E. (2015). How smart, connected products are transforming companies. *Harvard Business Review*, 93(10):96–112.
- Puget, J. F. (2015). Analytics maturity models. <https://t.co/7yYRw8a8EI>.
- Rana, K. and Vickson, R. (1991). Routing container ships using lagrangean relaxation and decomposition. *Transportation Science*, 25(3):201–214.



- Reinhardt, L. B. (2011). *Routing and scheduling problems*. PhD thesis, Technical University of Denmark.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374.
- Ronen, D. (1983). Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, 12(2):119–126.
- Ronen, D. (1993). Ship scheduling: The last decade. *European Journal of Operational Research*, 71(3):325–333.
- Ronen, D. (2011). The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society*, 62(1):211–216.
- Schmidt, M. and Schöbel, A. (2015). Timetabling with passenger routing. *OR Spectrum*, 37(1):75–97.
- Schmidt, M. E. (2014). *Integrating routing decisions in public transportation problems*. Springer.
- Schöbel, A. (2012). Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510.
- Shintani, K., Imai, A., Nishimura, E., and Papadimitriou, S. (2007). The container shipping network design problem with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 43(1):39–59.
- Ship&Bunker (2015). Maersk line lays up triple-e, cools over option for eight 14,000 teu newbuilds. <http://shipandbunker.com/news/world/142093-maersk-line-lays-up-triple-e-cools-over-option-for-eight-14000-teu-newbuilds>.
- Stahlbock, R. and Voß, S. (2008). Operations research at container terminals: a literature update. *Or Spectrum*, 30(1):1–52.
- Stopford, M. (2009). *Maritime Economics 3e*. Routledge.
- Toth, P. and Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications*, volume 18. SIAM.
- UNCTAD (2014). Review of maritime transport. Technical report, United Nations Conference on Trade and Development.
- Vis, I. F. and De Koster, R. (2003). Transshipment of containers at a container terminal: An overview. *European journal of operational research*, 147(1):1–16.

- Wang, S. and Meng, Q. (2012). Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):701–714.
- Wen, M., Ropke, S., Petersen, H., Larsen, R., and Madsen, O. (2015). Full-shipload tramp ship routing and scheduling with variable speeds. *Computers & Operations Research*.
- WSJ (2015). Maersk line ceo calls for container-shipping consolidation. <http://www.wsj.com/articles/maersk-line-ceo-calls-for-container-shipping-consolidation-1444382176>.



# Big Data Optimization in Maritime Logistics

with B.D. Brouer and David Pisinger<sup>1</sup>

---

## Abstract

Seaborne trade constitutes nearly 80% of the world trade by volume and is linked into almost every international supply chain. Efficient and competitive logistic solutions obtained through advanced planning will not only benefit the shipping companies, but will trickle down the supply chain to producers and consumers alike. Large scale maritime problems are found particularly within liner shipping due to the vast size of the network that global carriers operate. This chapter will introduce a selection of large scale planning problems within the liner shipping industry. We will focus on the solution techniques applied and show how strategic, tactical and operational problems can be addressed. We will discuss how large scale optimization methods can utilize special problem structures such as separable/independent sub-problems and give examples of advanced heuristics using divide-and-conquer paradigms, decomposition and mathematical programming within a large scale search framework. We conclude the chapter by discussing future challenges of large scale optimization within maritime shipping and the integration of predictive big data analysis combined with prescriptive optimization techniques.

---

<sup>1</sup>Brouer, B.D., Karsten, C.V., and Pisinger, D. (2015). Big data optimization in maritime logistics. Accepted in: *Big Data and Optimization*. Springer



**Figure 2.1:** Seaborne trade constitutes nearly 80% of the world trade by volume, and calls for the solution of several large scale optimization problems involving big data. *Picture: Maersk Line.*

## 2.1 Introduction

Modern container vessels can handle up to 20,000 twenty-foot equivalent units (TEU). The leading companies may operate a fleet of more than 500 vessels and transport more than 10,000,000 full containers annually that need to be scheduled through the network. There is a huge pressure to fill this capacity and utilize the efficiency benefits of the larger vessels but at the same time markets are volatile leading to ever changing conditions. Operating a liner shipping network is truly a big-data problem, demanding advanced decisions based on state-of-the art solution techniques. The digital footprint from all levels in the supply chain provides opportunities to use data that drive a new generation of faster, safer, cleaner, and more agile means of transportation. Efficient and competitive logistic solutions obtained through advanced planning will not only benefit the shipping companies, but will trickle down the supply chain to producers and consumers alike.

Maritime logistics companies encounter large scale planning problems at both the strategic, tactical, and operational level. These problems are usually treated separately due to complexity and practical considerations, but as will be seen in this chapter the decisions are not always independent and should not be treated as such. Large scale maritime problems are found both within transportation of bulk cargo, liquefied gasses and particularly within liner shipping

due to the vast size of the network that global carriers operate. In 2014 the busiest container terminal in the world, Port of Shanghai, had a throughput of more than 35,000,000 TEU according to Seatrade Global, which is also approximately the estimated number of containers in circulation globally. This chapter will focus on the planning problems faced by a global carrier operating a network of container vessels and show how decision support tools based on mathematical optimization techniques can guide the process of adapting a network to the current market.

At the strategic level carriers determine their fleet size and mix along with which markets to serve thus deciding the layout of their network. The network spanning the globe serving tens of thousands of customers leads to a gazillion possible configurations for operating a particular network. At the tactical level schedules for the individual services and the corresponding fleet deployment is determined, while the routing of containers through the physical transportation network, stowage of containers on the vessels, berthing of the vessels in ports, and disruption management due to e.g. bad weather or port delays is handled at the operational level. In general these problems can be treated separately, but as the layout of the network will affect e.g. the routing of the containers the problems are far from independent.

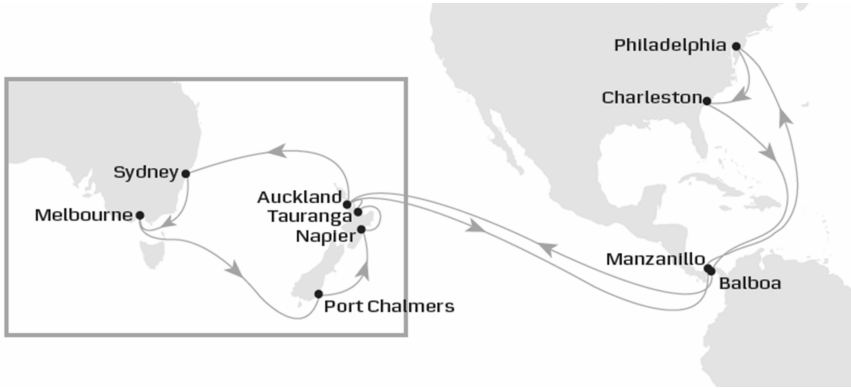
Operational data can lead to better predictions of what will happen in the future and carriers are constantly receiving sensor data from vessels that can help predict e.g. disruptions or required maintenance and similarly, data received from terminals can be used to predict delays and help vessels adjust sailing speed to save fuel. But given a predicted future scenario it may still not be obvious what the best actions are neither at the strategic, tactical or operational level. A large shipping company may be capable of producing good estimates of future demand and oil price fluctuations, or predicting possible disruptions. Under certain circumstances these predictions may require simple independent actions to adjust the network, but it is more likely that the actions will be dependent on other factors in the network. In that case difficult and complex here-and-now decisions must be made to adjust the transportation network optimally to the new situation. When there is a large number of decisions to be made and when the decisions influence each other prescriptive models based on optimization can help make the best choice. Predictive and prescriptive methods combined can serve as decision support tools and help select the best strategy, where the predictions made by machine learning algorithms, can be fed into large scale optimization algorithms to guide the decision process faced by carriers.

Most data in liner shipping are associated with some degree of uncertainty. First of all, demands are fluctuating over the year, and even if customers have booked a time slot for their containers these data are affected by significant uncertainty. In liner shipping no fees are paid if the customer is not delivering the booked number of containers, so customers may at any time choose to use another shipping company, or to postpone the delivery. This stimulates overbooking which adds uncertainty to the models. Port availabilities are also highly uncertain. If a vessel sticks to the normal time table, it can generally be assumed that the time slot is available, but if a vessel is delayed or the company wants to change the route, all port calls must be negotiated with the port authorities. This substantially complicates planning, and makes it necessary to use a trial and force method to find a good solution.

There are several different approaches for solving large scale optimization problems. If a problem exhibit a special separable structure it can be decomposed and solved more efficiently by using either column generation if the complication involves the number of variables or row generation if the number of constraints is too large (Barnhart et al., 1998; Bertsimas and Tsitsiklis, 1997; Costa, 2005; Desrosiers and Lübbecke, 2005), by dynamic programming (Cormen et al., 2001), or constraint programming (Rossi et al., 2006). For less structured or extremely large problems it can be advantageous to use (meta)-heuristics to obtain solutions quickly, but often of unknown quality (Burke and Kendall, 2005; Gendreau and Potvin, 2010). Finally it is frequently possible, with a good modeling of a problem, to rely solely on Linear Programming, LP, or Mixed Integer Programming, MIP, solvers, see e.g. Vielma (2015) for a discussion of modeling techniques and the trade-off between stronger versus smaller models. Algorithmic and hardware improvements have over the last three decades resulted in an estimated speed-up for commercial MIP solvers of a 200 billion factor (Bertsimas et al., 2014), making it feasible not only to solve large linear models but also more advanced integer decision models of realistic size. In practice a combination of the different techniques is often seen and maritime logistics gives an illustrative case of the importance of all of these large scale optimization methods.

## 2.2 Liner Shipping Network Design

The Liner Shipping Network Design Problem, LSNDP, is a core planning problem facing carriers. Given an estimate of the demands to be transported and a set of possible ports to serve, a carrier wants to design routes for its fleet of vessels and select which demands of containers to satisfy. A route, or *service*, is a set of similarly sized vessels sailing on a non-simple cyclic itinerary of ports according to a fixed, usually weekly, schedule. Hence the round trip duration for a vessel is assumed to be a multiple of a week and to ensure weekly frequency in the serviced ports a sufficient number of vessels is assigned. If a round trip of the vessel takes e.g. 6 weeks, then 6 vessels are deployed on the same route. To make schedules more robust buffer time is included to account for delays. However, delays may still lead to local speed increases which increases the overall energy consumption. An example of a service can be seen in Figure 2.2 which shows the Oceania-Americas Service with a round trip time of 10 weeks. The weekly departures may in some cases simplify the mathematical formulation of the problem, since customer demands and vessel capacities follow a weekly cycle. *Trunk services* serve central main ports and can be both inter and intra regional whereas *feeder services* serve a distinct market and typically visit one single main port and several smaller ports.



**Figure 2.2:** The Oceania-Americas Service (OC1). *Picture: Maersk Line.*

When the network has been determined the containers can be routed according to a fixed schedule with a predetermined trip duration. A given demand is loaded on to a service at its departure port, which may bring the demand directly to the destination port or the container can be unloaded at one or several intermediate ports for *transshipment* to another service before finally reaching its final destination. Therefore, the design of the set of services is complex, as



they interact through transshipments and the majority of containers are transshipped at least once during transport. A carrier aims for a network with high utilization, a low number of transshipments, and competitive transit times. Services are divided into a head- and a back-haul direction. The head haul direction is the most cargo intensive and vessels are almost full. Hence, the head haul generates the majority of the revenue and due to customer demand for fast delivery the head haul operates at increased speeds with nearly no buffer time for delays. The back haul operates at slower speeds with additional buffer time assigned. A delay incurred on the head haul is often recovered during the back-haul.

In practice a carrier will never re-design a network from scratch as there are significant costs associated with the reconfiguration (Tierney et al., 2013). Rather, the planners or network design algorithms will take the existing network and suggest incremental changes to adjust the network to the current economic environment. Most network changes requires evaluation of the full cargo routing problem to evaluate the quality of the network since regional changes can have unintended consequences in the entire network.

Routing of both vessels and containers are in most state-of-the-art methods considered simultaneously (Agarwal and Ergun, 2008; Álvarez, 2009; Brouer et al., 2014b, 2015; Reinhardt and Pisinger, 2012; Shintani et al., 2007), as these problems are completely interrelated. However, several of the before mentioned approaches exploit the fact that the problems are separable into two tiers and design algorithms utilizing this structure. The cargo routing reduces to a multicommodity flow problem, MCF, and serves as the lower tier where the revenue of the network is determined. The vessel routing problem reduces to a (more complex) problem of cycle generation and corresponds to the upper tier, where the cost of the network is determined. The following section gives insight to the container routing problem and its relation to the multi commodity flow problem.

## 2.2.1 Container Routing

We define  $G = (N, A)$  to be a directed graph with nodes  $N$  and edges  $A$ . The node set  $N$  represents the geographical locations in the model i.e. ports and the arc set  $A$  connects the ports. The arcs are determined by the scheduled itineraries and the cargo capacity is determined by the assignment of vessels to the schedule. Let  $K$  be the set of commodities to transport,  $q_k$  be the amount of commodity  $k \in K$  that is available for transport, and  $u_{ij}$  be the capacity of edge  $(i, j)$ . We assume that each commodity has a single origin node,  $O_k$ , and

a single destination node,  $D_k$ .

There are two commonly used formulations of the MCF based on either arc or path flow variables. The arc flow formulation can be stated as follows. For each node  $i \in N$  and commodity  $k \in K$  we define  $q(i, k) = q^k$  if  $i = O_k$ ,  $q(i, k) = -q^k$  if  $i = D_k$ , and  $q(i, k) = 0$  otherwise. For each node  $i \in N$  we define the set of edges with tail in node  $i$  as  $\delta^+(i) = \{(j, j') \in A : j = i\}$  and head in node  $i$  as  $\delta^-(i) = \{(j, j') \in A : j' = i\}$ .

With this notation the MCF problem can be stated as the following LP:

$$\min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k \quad (2.1)$$

$$\text{s.t.} \quad \sum_{(j,j') \in \delta^+(i)} x_{jj'}^k - \sum_{(j,j') \in \delta^-(i)} x_{jj'}^k = q(i, k) \quad i \in N, k \in K \quad (2.2)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} \quad (i, j) \in A \quad (2.3)$$

$$x_{ij}^k \geq 0 \quad (i, j) \in A, k \in K \quad (2.4)$$

The objective function (2.1) minimizes the cost of the flow. The flow conservation constraint (2.2) ensures that commodities originates and terminates in the right nodes. The capacity constraint (2.3) ensures that the capacity of each edge is respected. The formulation has  $|K||A|$  variables and  $|A| + |K||N|$  constraints. The number of variables is hence polynomially bounded, but for large graphs like the ones seen in global liner shipping networks this formulation requires excessive computation time and may even be too large for standard LP-solvers (see e.g. Brouer et al. (2011)).

The *block-angular structure* of the constraint matrix in the arc-flow formulation can be exploited and by *Dantzig-Wolfe decomposition* it is possible to get a reformulation with a master problem considering paths for all commodities, and a sub-problem defining the possible paths for each commodity  $k \in K$ . We note that in general any arc flow can be obtained as a convex combination of path flows. In the path-flow formulation each variable,  $f^p$ , in the model corresponds to a path,  $p$ , through the graph for a specific commodity. The variable states how many units of a specific commodity that is routed through the given path, the cost of each variable is given by the parameter  $c_p$ . Let  $P^k$  be the set of all feasible paths for commodity  $k$ ,  $P^k(a)$  be the set of paths for commodity  $k$  that uses edge  $a$  and  $P(a) = \cup_{k \in K} P^k(a)$  is the set of all paths that use edge  $a$ .

The model then becomes:

$$\min \sum_{k \in K} \sum_{p \in P^k} c_p f^p \quad (2.5)$$

$$\text{s.t.} \quad \sum_{p \in P^k} f^p = q_k \quad k \in K \quad (2.6)$$

$$\sum_{p \in P(a)} f^p \leq u_{ij} \quad (i, j) \in A \quad (2.7)$$

$$f^p \geq 0 \quad k \in K, p \in P^k \quad (2.8)$$

The objective function (2.5) again minimizes the cost of the flow. Constraint (2.6) ensures that the demand of each commodity is met and constraint (2.7) ensures that the capacity limit of each edge is obeyed. The path-flow model has  $|A| + |K|$  constraints, but the number of variables is, in general, growing exponentially with the size of the graph. However, using column generation the necessary variables can be generated dynamically and in practice the path-flow model can often be solved faster than the arc-flow model for large scale instances of the LSND problem (Brouer et al., 2011).

Column generation operates with a reduced version of the LP (2.5)-(2.8), which is called the master problem. The master problem is defined by a reduced set of columns  $Q^k \subseteq P^k$  for each commodity  $k$  such that a feasible solution to the LP (2.5)-(2.8) can be found using variables from  $\cup_{k \in K} Q^k$ . Solving this LP gives rise to dual variables  $\pi_k$  and  $\lambda_{ij}$  corresponding to constraint (2.6) and (2.7), respectively. For a variable  $j \in \cup_{k \in K} P^k$  we let  $\kappa(j)$  denote the commodity that a variable serves and let  $p(j)$  represent the path corresponding to the variable  $j$ , represented as the set of edges traversed by the path. Then we can calculate the *reduced cost*  $\bar{c}_j$  of each column  $j \in \cup_{k \in K} P^k$  as follows:

$$\bar{c}_j = \sum_{(i,j) \in p(j)} (c_{ij}^{\kappa(j)} - \lambda_{ij}) - \pi_{\kappa(j)}.$$

If we can find a variable  $j \in \cup_{k \in K} (P^k \setminus Q^k)$  such that  $\bar{c}_j < 0$  then this variable has the potential to improve the current LP solution and should be added to the master problem, which is resolved to give new dual values. If, on the other hand, we have that  $\bar{c}_j \geq 0$  for all  $j \in \cup_{k \in K} (P^k \setminus Q^k)$  then we know the master problem defined by  $Q^k$  provides the optimal solution to the complete problem (for more details see Karsten et al. (2015)). In order to find a variable with negative reduced cost or prove that no such variable exists we solve a sub-problem for each commodity. The sub-problem seeks the feasible path for commodity  $k$  with minimum reduced cost given the current dual values. Solving this problem

amounts to solving a shortest path problem from source to destination of the commodity with edge costs given by  $c_{ij} - \lambda_{ij}$  and subtracting  $\pi_k$  from this cost in order to get the reduced cost. As will be seen later we can extend the model to reject demands by including additional variables with an appropriate penalty. When solving the shortest path problem additional industry constraints such as number of transshipments, trade policies, or time limits on cargo trip duration can be included. Including such constraints will increase the complexity of the sub-problem as the resulting problem becomes a resource constrained shortest path problem. Karsten et al. (2015) has made a tailored algorithm for a cargo routing problem considering lead times and show that it does not necessarily increase the solution time to include transit time constraints, mainly because the size of solution space is reduced. Additionally, Karsten et al. (2015) give an overview of graph topologies accounting for transshipment operations when considering transit times.

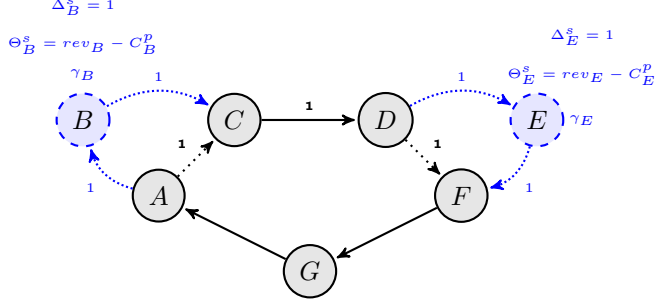
To construct routes used in the upper tier of the network design problem we will go through a more recent approach in the next section which use an advanced mathematical programming based heuristic to solve the problem within a large scale search framework. In general, when a generic network has been designed it is transformed into a physical sailing network by determining a specific schedule, deploying vessels from the available fleet and deciding on the speed and actual flow of containers. Some aspects of the tactical and operational decisions can of course be integrated in the network design process at the cost of computational tractability, but with the potential benefit of higher quality networks.

## 2.3 Matheuristic for Network Design

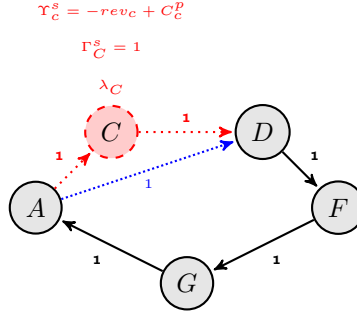
Mathematical programming models of the LSNDP are closely related to the capacitated fixed charge network design problem (Gendron et al., 1999) in installing a discrete set of capacities for the set of commodities  $K$ . However, the capacity installed must reflect the routing of container vessels according to the specification of a service as defined in the beginning of this section. Therefore, it is also related to pick-up and delivery vehicle routing problems (Toth and Vigo, 2014), however being significantly harder to solve as a consequence both of the non-simple cyclic routes, the multiple commodities and the vast size of real life networks. As a consequence optimal methods can only solve very insignificant instances of the LSNDP (Álvarez, 2009; Reinhardt and Pisinger, 2012) or provide lower bounds (Plum et al., 2014). Several algorithms for solving larger instances of the LSNDP can be categorized as *matheuristics* combining math-

emational programming with meta heuristics exploiting the two tier structure, where the variables of the upper tier describe a service and variables of the lower tier describe the container routing (for a reference model of the LSNDP see Brouer et al. (2014a)). Agarwal and Ergun (2008) apply a heuristic Benders decomposition algorithm as well as a Branch and Bound algorithm for heuristically generated routing variables, Álvarez (2009) applies a tabu search scheme, where the routing variables are generated by a mathematical program based on the dual values of the lower tier MCF problem in each iteration. Brouer et al. (2014a) use a heuristic column generation scheme, where the routing columns are generated by an integer program based on information from both tiers of the LSNDP along with a set of business rules. The integer program in Brouer et al. (2014a) constructs a single, (possibly non-simple) cyclic route for a given service configuration of vessel class and speed. Route construction is based on the Miller-Tucker-Zemlin subtour elimination constraints known from the CVRP to enumerate the port calls in a non-decreasing sequence. This makes high quality routings for smaller instances of the LSNDP, but for large scale instances it becomes necessary to select a small cluster of related ports in order to efficiently solve the integer program used in the heuristic. A different matheuristic approach is seen in Brouer et al. (2014b, 2015), where the core component in a large scale neighborhood search is an integer program designed to capture the complex interaction of the cargo allocation between routes. The solution of the integer program provides a set of moves in the composition of port calls and fleet deployment. Meta-heuristics for the LSNDP are challenged by the difficulty of predicting the changes in the multicommodity flow problem for a given move in the solution space without reevaluating the MCF at the lower tier. The approach of Brouer et al. (2014b) relies on estimation functions of changes in the flow and the fleet deployment related to inserting or removing a port call from a given service and network configuration. Flow changes and the resulting change in the revenue are estimated by solving a series of shortest path problems on the residual graph of the current network for relevant commodities to the insertion/removal of a port call along with an estimation of the change in the vessel related cost with the current fleet deployment.

(a) Blue nodes are evaluated for insertion corresponding to variables  $\gamma_i$  for the set of ports in the neighborhood  $N^s$  of service  $s$



(b) Red nodes are evaluated for removal corresponding to variables  $\lambda_i$  for the set of current port calls  $F^s$  on service  $s$ .



**Figure 2.3:** The estimation functions for insertion and removal of port calls.

Given a total estimated change in revenue of  $rev_i$  and port call cost of  $C_i^p$  Figure 2.3a illustrate estimation functions for the change in revenue ( $\Theta_i^s$ ) and duration ( $\Delta_i^s$ ) increase for inserting port  $i$  into service  $s$  controlled by the binary variable  $\gamma_i$ . The duration controls the number of vessels needed to maintain a weekly frequency of service. Figure 2.3b illustrate the estimation functions for the change in revenue ( $\Upsilon_i^s$ ) and decrease in duration ( $\Gamma_i^s$ ) for removing port  $i$  from service  $s$  controlled by the binary variable  $\lambda_i$ . Insertions/removals will affect the duration of the service in question and hence the needed fleet deployment modeled by the integer variable  $\omega_s$  representing the change in the number of vessels deployed. The integer program (2.9)-(2.16) expresses the neighborhood of a single service,  $s$ .

$$\max \quad \sum_{i \in N^s} \Theta_i \gamma_i + \sum_{i \in F^s} \Upsilon_i \lambda_i - C_V^{e(s)} \omega_s \quad (2.9)$$

$$\text{s.t.} \quad T_s + \sum_{i \in N^s} \Delta_i^s \gamma_i - \sum_{i \in F^s} \Gamma_i^s \lambda_i \leq 24 \cdot 7 \cdot (n_s^{e(s)} + \omega_s) \quad (2.10)$$

$$\omega_s \leq M_{e(s)} \quad (2.11)$$

$$\sum_{i \in N^s} \gamma_i \leq I_s \quad (2.12)$$

$$\sum_{i \in F^s} \lambda_i \leq R_s \quad (2.13)$$

$$\sum_{j \in L_i} \lambda_j \leq |L_i|(1 - \gamma_i) \quad i \in N^s \quad (2.14)$$

$$\sum_{j \in L_i} \lambda_j \leq |L_i|(1 - \lambda_i) \quad i \in F^s \quad (2.15)$$

$$\lambda_i \in \{0, 1\}, i \in F^s, \quad \gamma_i \in \{0, 1\}, i \in N^s, \quad \omega_s \in \mathbb{Z}. \quad (2.16)$$

The objective function (2.9) accounts for the expected change in revenue of the considered insertions and removals along with the weekly vessel cost  $C_V^{e(s)}$  of the vessel class  $e(s)$  deployed to service  $s$ . Constraint (2.10) considers the expected change in the duration of the service, where  $T_s$  is the current duration and  $n_s^{e(s)}$  is the number of vessels currently deployed to service  $s$ . The possible addition of vessels is bounded by the number of vessels available  $M_{e(s)}$  of type  $e$  in constraint (2.11). A limit on the number of insertions/removals respectively are introduced in constraints (2.12)-(2.13) to reduce the error of the estimation functions for multiple insertions/removals. The estimation functions also depend on the existing port calls for unloading the commodities introduced by the insertions as well as the ports used for rerouting commodities when removing ports. This is handled by introducing a lockset  $L_i$  for each insertion/removal expressed in constraints (2.14)-(2.15). The integer program is solved iteratively for each service in the current network and the resulting set of moves are evaluated for acceptance in a simulated annealing framework. The procedure is an improvement heuristic (Archetti and Speranza, 2014) fine tuning a given network configuration. The algorithm in its entirety constructs an initial network using a simple greedy construction heuristic. The improvement heuristic is applied as a move operator for intensification of the constructed solution. To diversify the solution a perturbation step is performed at every tenth loop through the entire set of services. The perturbation step alters the service composition in the network by removing entire services with low utilization and introducing a set of new services based on the greedy construction heuris-

tic for undeployed vessels. To evaluate the matheuristic the public benchmark suite, *Liner-lib*, for liner shipping network design problems is used.

## 2.4 Computational Results using *Liner-lib*

*Liner-lib* is a public benchmark suite for the LSNBP presented by Brouer et al. (2014a). The data instances of the benchmark suite are constructed from real-life data from the largest global liner-shipping company, Maersk Line, along with several industry and public stakeholders. *Liner-lib* consists of seven benchmark instances available at <http://www.linerlib.org> (see Brouer et al. (2014a) for details on the construction of the data instances). Each instance can be used in a low, medium, and high capacity case depending on the fleet of the instance. Table 2.1 presents some statistics on each instance ranging from smaller networks suitable for optimal methods to large scale instances spanning the globe. Currently published results are available for 6 of the 7 instances, leaving the *WorldLarge* instance unsolved.

Category	Instance and description	$ P $	$ K $	$ E $	min $v$	max $v$
Single-hub instances	<b>Baltic</b> Baltic Sea, Bremerhaven as hub	12	22	2	5	7
	<b>WAF</b> West Africa, Algeciras as hub	19	38	2	33	51
Multi-hub instance	<b>Mediterranean</b> Mediterranean Sea, Algeciras, Tangier, and Gioia Tauro as hubs	39	369	3	15	25
Trade-lane instances	<b>Pacific</b> Asia - US West Coast	45	722	4	81	119
	<b>AsiaEurope</b> Europe, Middle East, and Far East regions	111	4,000	6	140	212
World instances	<b>Small</b> 47 main ports worldwide identified by Maersk Line	47	1,764	6	209	317
	<b>Large</b> 197 ports worldwide identified by Maersk Line	197	9,630	6	401	601

**Table 2.1:** The instances of the benchmark suite with indication of the number of ports ( $|P|$ ), the number of origin-destination pairs ( $|K|$ ), the number of vessel classes ( $|E|$ ), the minimum (min  $v$ ) and maximum number of vessels (max  $v$ ).

*Liner-lib* contains data on ports including port call cost, cargo handling cost and draft restrictions, distances between ports considering draft and canal traversal, vessel related data for capacity, cost, speed interval and bunker con-



sumptions, and finally a commodity set with quantities, revenue, and maximal transit time. The commodity data reflects the current imbalance of world trade and the associated differentiated revenue. It is tailored for models of the LSNDP, but may provide useful data for related maritime transportation problems.

Computational results for *Liner-lib* are presented in Plum et al. (2014); Brouer et al. (2014a,b). Brouer et al. (2014a) presented the first results for the benchmark suite using the reference model (Brouer et al., 2014a) with biweekly frequencies for the feeder vessel classes and weekly frequencies for remaining classes. The heuristic column generation algorithm is used to solve all instances but the Large world instance with promising results. Brouer et al. (2014b) present computational results using the reference model with weekly frequencies for all vessel classes which has a more restricted solution space than Brouer et al. (2014a). As a consequence the solutions from Brouer et al. (2014b) are feasible for the model used in Brouer et al. (2014a), but not vice-versa. However, the computational results of Brouer et al. (2014b) indicate that the matheuristic using an improvement heuristic based on integer programming scales well for large instances and holds the current best known results for the Pacific, World Small and AsiaEurope instances. Plum et al. (2014) present a service flow model for the LSNDP using a commercial MIP solver presenting results for the two Baltic and WAF instances of *Liner-lib*. For details on the results the reader is referred to the respective papers. *Liner-lib* is currently used by researchers at a handful of different universities worldwide and may provide data for future results on models and algorithms for LSNDP.

## 2.5 Empty Container Repositioning

In extension of the network design process a liner shipping company must also consider revenue management at a more operational level. Requests for cargo can be rejected if it is not profitable to carry the containers, or if bottlenecks in the network make it infeasible. Moreover, empty containers tend to accumulate at importing regions due to a significant imbalance in world trade. Therefore, repositioning empty containers to exporting regions impose a large cost on liner shippers, and these costs need to be incorporated in the revenue model. Since larger shipping companies at any time have several millions of containers in circulation, these decisions are extremely complex and require advanced solution methods.

Álvarez (2009) presented a study of large scale instances of the liner service network design problem. The cargo allocation problem is solved as a sub-problem

of the tabu search algorithm solving the network design problem. Meng and Wang (2011) study a network design problem selecting among a set of candidate shipping lines while considering the container routing problem along with the repositioning of empty containers. The model is formulated as a minimum cost problem and as Erera et al. (2005) the model handle loaded end empty containers simultaneously, however it does not allow load rejection and only seek to minimize the cost of transport. Song and Dong (2012) consider a problem of joint cargo routing and empty container repositioning at the operational level accounting for the demurrage and inventory cost of empty containers. Like most other works on empty repositioning it is a cost minimizing problem where load rejection is not allowed.

Brouer et al. (2011) present a revenue management model for strategic planning within a liner shipping company. A mathematical model is presented for maximizing the profit of cargo transportation while considering the possible cost of repositioning empty containers.

The booking decision of a liner shipper considering empty container repositioning can be described as a specialized multi-commodity flow problem with inter-balancing constraints to control the flow of empty containers.

Similarly to the pure cargo routing problem we can define a commodity as the tuple  $(O_k, D_k, q_k, r_k)$  representing a demand of  $q_k$  in number of containers from node  $O_k$  to node  $D_k$  with a sales price per unit of  $r_k$ . The unit cost of arc  $(i, j)$  for commodity  $k$  is denoted  $c_{ij}^k$ . The non-negative integer variable  $x_{ij}^k$  is the flow of commodity  $k$  on arc  $(i, j)$ . The capacity of arc  $(i, j)$  is  $u_{ij}$ . To model the empty containers an empty super commodity  $k_e$  is introduced. The flow of the empty super commodity is defined for all  $(i, j) \in A$  as the integer variables  $x_{ij}^{k_e}$ . The unit cost of arc  $(i, j)$  for commodity  $k_e$  is denoted  $c_{ij}^{k_e}$ . The empty super commodity has no flow conservation constraints and appear in the objective with a cost and in the bundled capacity and inter-balancing constraints. For convenience the commodity set is split into the loaded commodities and the empty super commodity: Let  $K_F$  be the set of loaded commodities. Let  $K_e$  be the set of the single empty super commodity. Finally, let  $K = K_F \cup K_e$ . The inter-balancing constraints also introduce a new set of variables representing leased containers at a node. The cost of leasing is modeled in the objective. Let  $c_i^l$  be the cost of leasing a container at port  $i$ , while  $l_i$  is the integer leasing variable at port  $i$ . Demand may be rejected, due to capacity constraints and unprofitability from empty repositioning cost. The slack variable  $\gamma_k$  represents the amount of rejected demand for commodity  $k$ .

### 2.5.1 Path Flow Formulation

In the following we introduce a path flow model which is an extension of model (2.5)-(2.8). Again, let  $p$  be a path connecting  $O_k$  and  $D_k$  and  $P_k$  be the set of all paths belonging to commodity  $k$ . The flow on path  $p$  is denoted by the variable  $f^p$ . The binary coefficient  $a_{ij}^p$  is one if and only if arc  $(i, j)$  is on the path  $p$ . Finally,  $c_p^k = \sum_{(i,j) \in A} a_{ij}^p c_{ij}^k$  is the cost of path  $p$  for commodity  $k$ . The master problem is:

$$\max \sum_{k \in K_F} \sum_{p \in P_k} (r_k - c_p^k) f^p - \sum_{(i,j) \in A} c_{ij}^{k_e} x_{ij}^{k_e} - \sum_{i \in N} c_l^i l^i \quad (2.17)$$

$$\text{s.t.} \quad \sum_{k \in K_F} \sum_{p \in P_k} a_{ij}^p f^p + x_{ij}^{k_e} \leq u_{ij} \quad (i, j) \in A \quad (2.18)$$

$$\sum_{p \in P_k} f^p + \gamma_k = q_k \quad k \in K_F \quad (2.19)$$

$$\sum_{k \in K_F} \sum_{p \in P_k} \sum_{j \in N} (a_{ij}^p - a_{ji}^p) f^p + x_{ij}^{k_e} - x_{ji}^{k_e} - l^i \leq 0 \quad i \in N \quad (2.20)$$

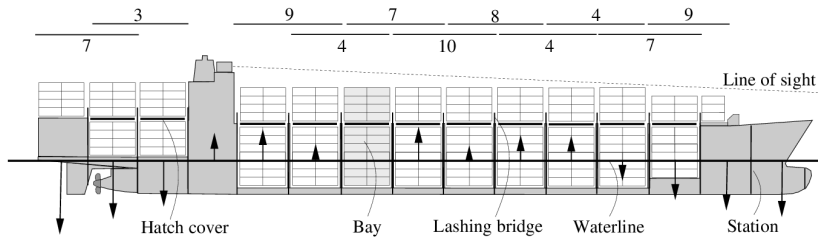
$$f^p \in \mathbb{Z}_+, \quad p \in P_k, \quad \gamma_k \in \mathbb{Z}_+, \quad k \in K_F \quad (2.21)$$

$$x_{ij}^{k_e} \in \mathbb{Z}_+, \quad (i, j) \in A, \quad l^i \in \mathbb{Z}_+, \quad i \in N \quad (2.22)$$

where the  $x_{ij}^{k_e}$  variables can be replaced by  $\sum_{p \in P_k} a_{ij}^p f^p$  for all  $k \in K_F$ . The convexity constraints for the individual sub-problems (2.19) bound the flow between the  $(O_k, D_k)$  pair from above (a maximal flow of  $q_k$  is possible).

Paths are generated on the fly using delayed column generation. Brouer et al. (2011) report computational results for eight instances based on real life shipping networks, showing that the delayed column generation algorithm for the path flow model clearly outperforms solving the arc flow model with the CPLEX barrier solver. In order to fairly compare the arc and path flow formulation a basic column generation algorithm is used for the path flow model versus a standard solver for the arc flow model. Instances with up to 234 ports and 293 vessels for 9 periods were solved in less than 35 minutes with the column generation algorithm. The largest instance solved for 12 periods contains 151 ports and 222 vessels and was solved in less than 75 minutes.

The algorithm solves instances with up to 16,000 commodities over a twelve month planning period within one hour. Integer solutions are found by simply rounding the LP solution. The model of Erera et al. Erera et al. (2005) is solved to integer optimality using standard solvers as opposed to the rounded integer solution presented here. The problem sizes of Brouer et al. (2011) are



**Figure 2.4:** The arrangement of bays in a small container vessel, and stacking heights. The arrows indicate forces. *Picture: Pacino (2012).*

significantly larger than those of Erera et al. (2005) and the rounded integer solutions lead to a gap of at most 0.01% from the LP upper bound of the path flow formulation, which is very acceptable, and far below the level of uncertainty in the data. The results of Erera et al. (2005) confirm the economic rationale in simultaneously considering loaded and empty containers.

## 2.6 Container Vessel Stowage Plans

With vessels carrying up to 20,000 TEU, stowage of the containers on board is a non-trivial task demanding fast algorithms as the final load list is known very late. Stowage planning can be split into a master planning problem and a more detailed slot planning problem. The master planning problem should decide a proper mixture of containers, so that constraints on volume, weight, and reefer plugs are respected. The slot planning problem should assign containers to slots in the vessel so that the loading and unloading time in ports can be minimized. The vessel must be seaworthy, meaning that stability and stress constraints must be respected.

Figure 2.4 illustrates the arrangement of bays in a container vessel. Containers are loaded bottom-up in each bay up to a given stacking height limited by the line of sight and other factors. Some containers are loaded below deck, while other containers are loaded above the hatch cover. The overall weight sum of containers may not exceed a given limit, and the weight need to be balanced. Moreover, torsions should be limited, making it illegal to e.g. only load containers at the same front and end of the vessel. Refrigerated containers (reefers) need to be attached to an electric plug. Only a limited number of plugs are available, and these plugs are at specific positions.

A good stowage plan should make sure that it is not necessary to rearrange

containers at each port call. All containers for the given port should be directly accessible when arriving to the port, and there should be sufficient free capacity for loading new containers. If several cranes are available in a port, it is necessary to ensure that all cranes can operate at the same time without blocking for each other.

Pacino (2012) presents a MIP model for the master problem. The model is based on Pacino et al. (2011); Pacino and Jensen (2012). The model considers both 20' and 40' containers, assuming that two 20' containers can fit in the slot of a 40' container provided that the middle is properly supported. Four types of containers are considered: light, heavy, light reefer, and heavy reefer. Decision variables are introduced for each slot, indicating how many of each container type will be loaded in the slot.

The MIP model has a large number of constraints: First of all, a load list and cargo estimates are used to calculate the number of containers of each type that needs to be stowed. Moreover, every slot has a capacity of dry containers and reefers. An overall weight limit given by the capacity of the vessel is also imposed. When calculating the weight limit, average values for light and heavy containers are used to ease the calculations. Trim, draft, buoyancy, and stability are calculated as a function of displacement and center of gravity of the vessel. Finally, a number of penalties associated with a given loading are calculated. These include hatch-overstowage, overstowage in slots, time needed for loading, and excess of reefer containers. The objective of the model minimizes a weighted sum of the penalties.

Pacino (2012) show that the master planning problem is NP-hard. Computational results are reported for instances with vessel capacity up to around 10,000 TEU, visiting up to 12 ports involving more than 25,000 lifts (crane moves of a container). Several of these instances can be solved within 5 minutes up to a 5% gap, using a MIP-solver.

### 2.6.1 Mathematical Model

In the slot planning phase, the master plan is refined by assigning the containers to specific slots on board the vessel (Pacino and Jensen, 2013). This problem involves handling of a number of stacking rules, as well as constraints on stack heights and stack weight. Since several of the containers are already stowed on board the vessel the objective is to arrange containers with the same destination port in the same stack, free as many stacks as possible, minimize overstowage, and minimize the number of non-reefer containers assigned to reefer slots. Due to the large number of logical constraints in this problem Delgado et al. (2012) proposed a logical model using the following notation.  $\mathcal{S}$  is the set of stacks,  $\mathcal{T}_s$

is the set of tiers for stack  $s$ ,  $\mathcal{P}$  represents the aft ( $p = 1$ ) and fore ( $p = 2$ ) of a cell,  $\mathcal{C}$  is the set of containers to stow in the location and  $\mathcal{C}^P \subset \mathcal{C}$  is the subset of containers in the release, i.e. the set of containers that are already on-board the vessel.  $x_{stp} \in \mathcal{C} \cup \{\perp\}$  is a decision variable indicating the location of a container  $c \in \mathcal{C}$  or the empty assignment  $\perp$ .  $A_{stp}^{40}$  is a binary variable indicating if the cell in stack  $s$ , tier  $t$ , and position  $p$  can hold a 40' foot container and similarly  $A_{stp}^{20}$  is one if a slot can hold a 20' container.  $A_{stp}^R$  is a binary indicator for the position of reefer plugs.  $W_s$  and  $H_s$  is the maximum weight and height of stack  $s$ . The attribute functions use  $w(c)$  and  $h(c)$  for the weight and height of a container.  $r(c)$  is true iff the container is a reefer,  $\perp(c)$  is true iff  $c = \perp$ ,  $f(c)$  is true iff the container is 40', and  $t(c)$  is true iff it is a 20' container. Then the logical model is:

$$|\{x_{stp} = c | s \in \mathcal{S}, t \in \mathcal{T}_s, p \in \mathcal{P}\}| = 1 \quad c \in \mathcal{C} \quad (2.23)$$

$$x_{s_c t_c p_c} = c \quad c \in \mathcal{C}^P \quad (2.24)$$

$$\neg f(x_{st1}) \wedge (f(x_{st2}) \implies \perp(x_{st1})) \quad s \in \mathcal{S}, t \in \mathcal{T}_s \quad (2.25)$$

$$t(x_{stp}) \implies A_{stp}^{20} \quad s \in \mathcal{S}, t \in \mathcal{T}_s, p \in \mathcal{P} \quad (2.26)$$

$$f(x_{st1}) \implies A_{st}^{40} \quad s \in \mathcal{S}, t \in \mathcal{T}_s \quad (2.27)$$

$$\sum_{t \in \mathcal{T}_s} (w(x_{st1}) + w(x_{st2})) \leq W_s \quad s \in \mathcal{S} \quad (2.28)$$

$$\sum_{t \in \mathcal{T}_s} \max(h(x_{st1}), h(x_{st2})) \leq H_s \quad s \in \mathcal{S} \quad (2.29)$$

$$\neg \perp(x_{stp}) \implies (t(x_{s(t-1)1}) \wedge t(x_{s(t-1)2})) \vee f(x_{s(t-1)1}) \quad s \in \mathcal{S}, t \in \mathcal{T}_s \setminus \{1\}, p \in \mathcal{P} \quad (2.30)$$

$$f(x_{st1}) \implies \perp t(x_{s(t+1)p}) \quad s \in \mathcal{S}, t \in \mathcal{T}_s \setminus \{N_s^T\}, p \in \mathcal{P} \quad (2.31)$$

$$r(x_{stp}) \wedge t(x_{stp}) \implies A_{stp}^R \quad s \in \mathcal{S}, t \in \mathcal{T}_s, p \in \mathcal{P} \quad (2.32)$$

$$r(x_{st1}) \wedge f(x_{st1}) \implies A_{st1}^R \vee A_{st2}^R \quad s \in \mathcal{S}, t \in \mathcal{T}_s \quad (2.33)$$

Constraints (2.23)-(2.24) ensure that each container is assigned to exactly one slot. Constraint (2.25) ensures that a 40' container occupies both the aft and fore position of a cell. The assignments need to respect cell capacity (2.26)-(2.27), stack height and stack weight limits (2.28)-(2.29). Two 20' containers can be stowed in a 40' slot, if properly supported from below (2.30). This means that 40' container can be stacked on top of two 20' containers, but not the other way around (2.31). Reefer containers need to be assigned to slots with a power plug (2.32)-(2.33).

In order to minimize the objective function Delgado et al. (2012) propose to use Constraint-Based Local Search. The framework combines local search al-

gorithms with constraint programming. The constraint satisfaction part of the problem is transformed to an optimization problem where the objective is to minimize constraint violation. A hill-climbing method is used to optimize the slot planning. The neighborhood in the search consists of swapping containers between a pair of cells.

Pacino (2012) report computational results for 133 real-life instances, showing that the local search algorithm actually finds the optimal solution in 86% of the cases. The running times are below 1 second.

## 2.7 Bunker Purchasing

In a liner shipping network bunker fuel constitutes a very large part of the variable operating cost for the vessels. Also, the inventory holding costs of the bunker on board may constitute a significant expense to the liner shipping company. Bunker prices are fluctuating and generally correlated with the crude oil price, but there are significant price differences between ports. This creates the need for frequent (daily) re-optimization of the bunker plan for a vessel, to ensure the lowest bunker costs. Bunker can be purchased on the spot market when arriving to a port, but normally it is purchased some weeks ahead of arrival. Long-term contracts between a liner shipping company and a port can result in reduced bunkering costs by committing the company to purchase a given amount of bunker. Bunkering contracts may cover several vessels sailing on different services, making the planning quite complex. The bunker purchasing problem is to satisfy the vessels consumption by purchasing bunkers at the minimum overall cost, while considering reserve requirements, and other operational constraints. Bunker purchasing problems involve big data. Real-life instances may involve more than 500 vessels, 40,000 port calls, and 750 contracts. For a vessel sailing on a given port to port voyage at a given speed, the bunker consumption can be fairly accurately predicted. This gives an advantage in bunker purchasing, when a vessel has a stable schedule known for some months ahead. The regularity in the vessel schedules in liner shipping allows for detailed planning of a single vessel.

Besbes and Savin (2009) consider different re-fueling policies for liner vessels and present some interesting considerations on the modeling of stochastic bunker prices using Markov processes. This is used to show that the bunkering problem in liner shipping can be seen as a stochastic capacitated inventory management problem. Capacity is the only considered operational constraint. More recently Wang and Meng (2015) examined re-fueling under a worst-case bunker consumption scenario.

The work of Plum and Jensen (2007) considers multiple tanks in the vessel and stochasticity of both prices and consumption, as well as a range of operational constraints. Yao et al. (2012) does not consider stochastic elements nor tanks, but has vessel speed as a variable of the model. The work of Kim et al. (2012) minimizes bunker costs as well as startup costs and inventory costs for a single liner shipping vessel. This is done by choosing bunker ports and bunker volumes but also having vessel round trip speed (and thus the number of vessels on the service) as a variable of the model. In Sheng et al. (2014) a model is developed which considers the uncertainty of bunker prices and bunker consumption, modeling their uncertainty by markov processes in a scenario tree. The work can be seen as an extension of (Yao et al., 2012), as it considers vessel speed as a variable within the same time window bounds. Capacity and fixed bunkering costs is considered, as is the holding / tied capital cost of the bunkers.

The studies described above do not consider bunker contracts, and all model the bunker purchasing for a single vessel.

### 2.7.1 Bunker Purchasing with Contracts

Plum et al. (2015) presented a decomposition algorithm for the Bunker Purchasing with Contracts Problem, BPCP, and showed that the model is able to solve even very large real-life instances. The model is based on writing up all bunkering patterns, and hence may be of exponential size. Let  $I$  be the set of ports visited on an itinerary,  $B$  be the set of bunker types, and  $V$  be the set of vessels. A contract  $c \in C$  has a minimal  $\underline{q}_c$  and maximal  $\bar{q}_c$  quantity that needs to be purchased. A contract  $c$  will give rise to a number of purchase options  $m \in M$ , i.e. discrete events where a specific vessel  $v$  calls a port within the time interval of a contract  $c$ , allowing it to purchase bunker at the specific price  $p_m$ . Each time a purchase is done at port  $i$  a startup cost  $sc_i$  is paid. Let  $R_v$  be the set of all feasible bunkering patterns for a vessel  $v$ . A bunkering pattern is feasible if a sufficient amount of bunker is available for each itinerary, including reserves. Bunker is available in various grades, and it is allowed to substitute a lower grade with a higher grade. In some areas, only low-sulphur bunker may be used, and this needs to be respected by the bunkering plan. Moreover initial and terminal criteria for bunker volumes must be met. Finding a legal bunkering pattern can be formulated as a MIP model (Plum et al., 2015) and solved by commercial solvers. Each pattern  $r \in R_v$  is denoted as a set of bunkerings.

Let  $u_r = \sum_{m \in M} (p_m l_m) + \sum_{i \in I} \sum_{v \in V} \sum_{b \in B} (\delta_{i,b} sc_i)$  be the cost for pattern



$r \in R_v$ . In this expression,  $l_m$  is the purchase of bunker for each purchase option  $m$ . and  $p_m$  is the price of option  $m$ . The binary variable  $\delta_{i,b}$  is set to one iff a purchase of bunker type  $b$  is made at port call  $i$ . Let  $\lambda_r$  be a binary variable, set to 1 iff the bunkering pattern  $r$  is used. Let  $o_{r,c}$  be the quantity purchased of contract  $c$  by pattern  $r$ . The BPCP can then be formulated as:

$$\min \sum_{v \in V} \sum_{r \in R_v} \lambda_r u_r + \sum_{c \in C} (\underline{s}_c \underline{w} + \bar{s}_c \bar{w}) \quad (2.34)$$

$$\text{s.t.} \quad \underline{q}_c - \underline{s}_c \leq \sum_{v \in V} \sum_{r \in R_v} \lambda_r o_{r,c} \leq \bar{q}_c + \bar{s}_c \quad c \in C \quad (2.35)$$

$$\sum_{r \in R_v} \lambda_r = 1 \quad v \in V \quad (2.36)$$

$$\lambda_r \in \{0, 1\} \quad r \in R_v \quad (2.37)$$

The objective minimizes the costs of purchased bunker, startup costs and slack costs. The parameters  $\underline{w}$  and  $\bar{w}$  denote a penalty for violating the minimal  $\underline{q}_c$  and maximal  $\bar{q}_c$  quantity imposed by contract  $c$ . Constraints (2.35) ensures that all contracts are fulfilled. Convexity constraints (2.36) ensure that exactly one bunker pattern is chosen for each vessel.

Due to the large number of columns in the model Plum et al. (2015) proposed to solve the LP relaxed model by Column Generation. Using the generated columns from the LP-solution, the resulting problem is solved to integer optimality using a MIP solver, leading to a heuristic solution for the original problem. Initially all dual variables are set to zero, a sub-problem is constructed for each vessel and solved as a MIP problem. The first master problem is then constructed with one solution for each vessel as columns. This master is solved and the first values are found. The sub-problems are resolved for all vessels (only the objective coefficients for the contracts needs updating) and new columns are generated for the master. This continues until no negative reduced cost columns can be generated, and the LP optimal solution is achieved. The sub-problems do not need to be solved to optimality since any column with negative reduced cost will ensure progress of the algorithm. Therefore the solver is allowed to return solutions to the sub-problem having a considerable optimality gaps. As the algorithm progresses, the allowable sub-problem gap is reduced.

A simple form of dual stabilization has been used in the implementation by (Plum et al., 2015) to speed up convergence. The Box-step method imposes a

box around the dual variables, which are limited from changing more than  $\pi_{max}$  per iteration. This has been motivated by the dual variables only taking on values  $\{-\underline{w}, \overline{w}, 0\}$  in the first iteration, these then stabilize at smaller numerical values in subsequent iterations. The model is able to solve even very large real-life instances involving more than 500 vessels, 40,000 port calls, and 750 contracts. First, column generation is used to solve the linearized model, and then a MIP solver is used to find an integer solution only using the generated columns. This results in a small gap in the optimal solution compared to if all columns were known. However, computational results show that the gap is never more than around 0.5% even for the largest instances. In practice the resulting gap of the algorithm, can be much smaller since the found solutions are benchmarked against a lower bound and not against the optimal solution.

An interesting side product of the model is the dual variables  $\underline{\pi}_c$  and  $\overline{\pi}_c$  for the upper and lower contract constraints (2.35). These values can be used to evaluate the gain of a given contract, which may be valuable information when (re)negotiating contracts.

Since bunker prices are stochastic of nature, future research should be focused on modeling the price fluctuation. However, the models tend to become quite complex and difficult to solve as observed by Plum and Jensen (2007), while only adding small extra improvements to the results. So a trade-off must be done between model complexity and gain in bunker costs. The work of Sheng et al. (2014) shows some promising developments in this important direction. Also, instruments from finance (bunker future or forward contracts, fixed price bunker fuel swaps) could be used to control risk in bunker purchasing, and to increase the margins on oil trade. Bunker purchasing for liner ships constitutes such a big market that it deserves a professional trading approach.

## 2.8 The Vessel Schedule Recovery Problem

It is estimated that approximately 70-80% of vessel round trips experience delays in at least one port. The common causes are bad weather, strikes in ports, congestions in passageways and ports, and mechanical failures. Currently, when a disruption occur, the operator at the shipping companies manually decides what action to take. For a single delayed vessel a simple approach could be to speed up. However, the consumption of bunker fuel is close to a cubic function of speed and vessels' speeds are limited between a lower and upper limit. So even though an expensive speed increase strategy is chosen, a vessel can arrive late for connections, propagating delays to other parts of the network. Having more than 10,000 containers on board a large vessel, calculating the overall consequences of re-routing/delaying these containers demands algorithms for big data. Disruption management is well studied within the airline industry (see Ball et al. (2007) or Clausen et al. (2010) for a review) and the network design of airlines resemble liner shipping networks inspiring the few works on disruption management found for liner shipping. Mulder et al. (2012) presents a markov decision model to determine the optimal recovery policy. The core idea is to reallocate buffer time within a schedule in order to recover from disruptions. Brouer et al. (2013) present the Vessel Schedule Recovery Problem (VSRP) handling a disruption in a liner shipping network by omitting port calls, swapping port calls or speeding up vessels in a predefined disruption scenario. The model and method will be presented in the following section.

### 2.8.1 Definitions

A given disruption scenario can mathematically be described by a set of vessels  $V$ , a set of ports  $P$ , and a time horizon consisting of discrete time slots  $t \in T$ . The time slots are discretized on port basis as terminal crews handling the cargo operate in shifts, which are paid for in full, even if arriving in the middle of a shift. Hence we only allow vessels arriving at the beginning of shifts. Reducing the graph to timeslots based on these shifts, also has the advantage of reducing the graph size, although this is a minor simplification of the problem. For each vessel  $v \in V$ , the current location and a planned schedule consisting of an ordered set of port calls  $H_v \subseteq P$  are known within the recovery horizon, a port call  $A$  can precede a port call  $B$ ,  $A < B$  in  $H_v$ . A set of possible sailings, i.e. directed edges,  $L_h$  are said to *cover* a port call  $h \in H_v$ . Each  $L_h$  represent a sailing with a different speed.

The recovery horizon,  $T$ , is an input to the model given by the user, based on the disruption in question. Inter-continental services will often recover by

speeding during ocean crossing, making the arrival at first port after an ocean crossing a good horizon, severe disruptions might require two ocean crossings. Feeders recovering at arrival to their hub port call would save many missed transshipments giving an obvious horizon. In combination with a limited geographical dimension this ensures that the disruption does not spread to the entire network.

The disruption scenario includes a set of container groups  $C$  with planned transportation scenarios on the schedules of  $V$ . A feasible solution to an instance of the VSRP is to find a sailing for each  $v \in V$  starting at the current position of  $v$  and ending on the planned schedule no later than the time of the recovery horizon. The solution must respect the minimum and maximum speed of the vessel and the constraints defined regarding ports allowed for omission or port call swaps. The optimal solution is the feasible solution of minimum cost, when considering the cost of sailing in terms of bunker and port fees along with a strategic penalty on container groups not delivered “on-time” or misconnecting altogether.

### 2.8.2 Mathematical Model

Brouer et al. (2013) use a time space graph as the underlying network, but reformulate the model to address the set of available recovery techniques, which are applicable to the VSRP.

The binary variables  $x_e$  for each edge  $e \in E_s$  are set to 1 iff the edge is sailed in the solution. Binary variables  $z_h$  for each port call  $h \in H_v$   $v \in V$  are set to 1 iff call  $h$  is omitted. For each container group  $c$  we define binary variables  $o_c \in \{0, 1\}$  to indicate whether the container group is delayed or not and  $y_c$  to account for container groups misconnecting. The parameter  $O_e^c \in \{0, 1\}$  is 1 iff container group  $c \in C$  is delayed when arriving by edge  $e \in L_{T_c}$ .  $B_c \in H_v$  is defined as the origin port for a container group  $c \in C$  and the port call where vessel  $v$  picks up the container group. Similarly, we define  $T_c \in H_w$  as the destination port for container group  $c \in C$  and the port call where vessel  $w$  delivers the container group. Intermediate planned transshipment points for each container group  $c \in C$  are defined by the ordered set  $I_c = (I_c^1, \dots, I_c^m)$ . Here  $I_c^i = (h_v^i, h_w^i) \in (H_v, H_w)$  is a pair of calls for different vessels ( $v, w \in V | v \neq w$ ) constituting a transshipment. Each container group  $c$  has  $m^c$  transshipments.  $M_c^e$  is the set of all non-connecting edges of  $e \in L_h$  that result in miss-connection of container group  $c \in C$ .  $M_c \in \mathbb{Z}_+$  is an upper bound on the number of transshipments for container group  $c \in C$ .

Let the demand of vessels  $v$  in a node  $n$  be given by  $S_v^n = -1$  if  $n = n_s^v$ ,  $S_v^n = 1$  if  $n = n_t^v$ , while  $S_v^n = 0$  for all other nodes. Then we get the following model:

$$\min \sum_{v \in V} \sum_{h \in H_v} \sum_{e \in L_h} c_e^v x_e + \sum_{c \in C} (c_c^m y_c + c_c^d o_c) \quad (2.38)$$

$$\text{s.t.} \quad \sum_{e \in L_h} x_e + z_h = 1 \quad v \in V, h \in H_v \quad (2.39)$$

$$\sum_{e \in n^-} x_e - \sum_{e \in n^+} x_e = S_v^n \quad v \in V, n \in N_v \quad (2.40)$$

$$y_c \leq o_c \quad c \in C \quad (2.41)$$

$$\sum_{e \in L_{T_c}} O_e^c x_e \leq o_c \quad c \in C \quad (2.42)$$

$$z_h \leq y_c \quad c \in C, h \in B_c \cup I_c \cup T_c \quad (2.43)$$

$$x_e + \sum_{\lambda \in M_c^e} x_\lambda \leq 1 + y_c \quad c \in C, e \in \{L_h | h \in B_c \cup I_c \cup T_c\} \quad (2.44)$$

$$x_e \in \{0, 1\} \quad e \in E_s$$

$$y_c, o_c \in \mathbb{R}_+ \quad c \in C$$

$$z_h \in \mathbb{R}_+ \quad v \in V, h \in H_v$$

The objective function (2.38) minimizes the cost of operating vessels at the given speeds, the port calls performed along with the penalties incurred from delaying or misconnecting cargo.

Constraints (2.39) are set-partitioning constraints ensuring that each scheduled port call for each vessel is either called by some sailing or omitted. The next constraints (2.40) are flow-conservation constraints. Combined with the binary domain of variables  $x_e$  and  $z_h$  they define feasible vessel flows through the time-space network. A misconnection is by definition also a delay of a container group and hence the misconnection penalty is added to the delay penalty, as formulated in (2.41). Constraints (2.42) ensure that  $o_c$  takes the value 1 iff container group  $c$  is delayed when arriving via the sailing represented by edge  $e \in E_s$ . Constraints (2.43) ensure that if a port call is omitted, which had a planned (un)load of container group  $c \in C$ , the container group is misconnected. Constraints (2.44) are coherence constraints ensuring the detection of container groups' miss-connections due to late arrivals in transshipment ports. On the left-hand side the decision variable corresponding to a given sailing,  $x_e$ , is added to the sum of all decision variables corresponding to having onward sailing resulting in miss-connections,  $\lambda \in M_c^e$ .

In (Brouer et al., 2013) the model has been tested on a number of real-life cases, including a delayed vessel, a port closure, a berth prioritization, and expected congestion. An analysis of the four real life cases, show that a disruption allowing to omit a port call or swap port calls may ensure timely delivery of cargo without having to increase speed and hence, a decision support tool based on the VSRP may aid in decreasing the number of delays in a liner shipping network, while maintaining a slow steaming policy. To operationalize this the rerouting of the actual flow and adjustment of the actual schedule must be incorporated in a real time system to enable here-and-now decisions. This is especially challenging for larger disruption scenarios than the ones described as the size of the problem grows exponentially.

## 2.9 Conclusions and Future Challenges

Maritime logistics companies operate in an environment which requires them to become more and more analytical. In general there are several insights to be gained from the data companies has available. Especially when companies start to use the forward looking analytical techniques rather than only using data for backward looking analysis (descriptive and diagnostic models) companies can unlock significant value from the collected data as shown in this chapter. Forward looking techniques (predictive models) can provide input for the decision making process where the best possible action is sought (prescriptive models). A pressing challenge in big data analysis today lies in the integration of predictive and prescriptive methods which combined can serve as valuable decision support tools. This chapter introduced a selection of large scale planning problems within maritime logistics with a primary focus on challenges found in the liner shipping industry. Focus has been on addressing strategic, tactical and operational problems by modern large scale optimization methods. However optimization within maritime logistics is complicated by the uncertainty and difficult accessibility of data. Most demands are only estimates, and for historic reasons even contracted cargo can be unreliable since there are no penalties associated with no-show cargo. To limit these uncertainties predictive machine learning techniques is an important tool. In particular, seasonal variations and similar trends can be predicted quite well and decision support systems should take such uncertainties into account. This can be done either by developing models where it is possible to re-optimize the problem quickly in order to meet new goals and use them interactively for decision support and for evaluating what-if scenarios suggested by a planner as there are still many decisions that will not be data-driven. Quantitative data can not always predict the future well in situations of e.g. one-time

events and generally extrapolation is hard. But in situations where we operate in an environment where data can be interpolated mathematical models may serve as great decision support tools by integrating the predictive models directly in the prescriptive model. With the large volume of data generated by carriers, increased quality of forecasts, and algorithmic improvements it may also be beneficial and even tractable to include the uncertainties directly in the decision models. A relatively new way of handling data uncertainty is by introducing uncertainty sets in the definition of the data used for solving large-scale LP's. The standard LP found as a sub-problem in many of the described problems can generically be stated as  $\min_x \{c^T x : Ax \leq b\}$ , where  $A$ ,  $b$ , and  $c$  contain the data of the problem at hand. As described previously in this chapter most of the data is associated with uncertainties but in *Robust Optimization* this can be handled by replacing the original LP with an uncertain LP  $\{\min_x \{c^T x : Ax \leq b\} : (A, b, c) \in \mathcal{U}\}$ . The best robust solution to the problem can be found by solving the Robust Counterpart of the problem, which is an semi-infinite LP  $\min_{x,t} \{t : c^T x \leq t, Ax \leq b \forall (A, b, c) \in \mathcal{U}\}$ . Clearly this LP is larger than the original LP, but with good estimates of the uncertainty sets the size can be manageable, further details can be found in Ben-Tal and Nemirovski (2002). As the accuracy of predictive models increases it will be possible to come up with good estimates for the uncertainty sets and thereby actually making it feasible to solve robust versions of the planning problems. In the MIP case the problems usually become much harder and often intractable with a few exceptions. An alternative approach to Robust Optimization is to handle the uncertainties via probability distributions on the data and use *Stochastic Programming* and solve the chance constrained program  $\min_{x,t} \{t : \text{Prob}_{(A,b,c) \sim P} \{c^T x \leq t, Ax \leq b\} \geq 1 - \epsilon\}$  or a two-stage stochastic program based on a set of scenarios. Again, machine learning algorithms can provide good estimates of the actual underlying distributions or expected scenarios and it may be possible to obtain results that are less conservative than the worst-case results provided by Robust Optimization, but the process can be more computationally extensive.

## Bibliography

- Agarwal, R. and Ergun (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42:175–196.
- Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11:186–208.
- Archetti, C. and Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2:223–246.

- Ball, M., Barnhart, C., Nemhauser, G., and Odoni, A. (2007). *Handbooks in Operations Research & Management Science: Transportation*, volume 14, chapter 1: Air Transportation - Irregular Operations and Control. Elsevier.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329.
- Ben-Tal, A. and Nemirovski, A. (2002). Robust optimization—methodology and applications. *Mathematical Programming*, 92(3):453–480.
- Bertsimas, D., King, A., and Mazumder, R. (2014). Best subset selection via a modern optimization lens. *Submitted to Annals of Statistics*.
- Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA.
- Besbes, O. and Savin, S. (2009). Going bunkers: The joint route selection and refueling problem. *Manufacturing and Service Operations Management*, 11:694–711.
- Brouer, B., Álvarez, J., Plum, C., Pisinger, D., and Sigurd, M. (2014a). A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48:281–312.
- Brouer, B., Desaulniers, G., Karsten, C., and Pisinger, D. (2015). A matheuristic for the liner shipping network design problem with transit time restrictions. *Lecture Notes in Computer Science*, Forthcoming.
- Brouer, B., Desaulniers, G., and Pisinger, D. (2014b). A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59.
- Brouer, B., Dirksen, J., Pisinger, D., Plum, C., and Vaaben, B. (2013). The vessel schedule recovery problem (VSRP) a MIP model for handling disruptions in liner shipping. *European Journal of Operational Research*, 224:362–374.
- Brouer, B., Pisinger, D., and Spoorendonk, S. (2011). Liner shipping cargo allocation with repositioning of empty containers. *INFOR: Information Systems and Operational Research*, 49:109–124.
- Burke, E. K. and Kendall, G. (2005). *Search methodologies*. Springer.
- Clausen, J., Larsen, A., Larsen, J., and Rezanova, N. J. (2010). Disruption management in the airline industry-concepts, models and methods. *Computers and Operations Research*, 37(5):809–821.



- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., et al. (2001). *Introduction to algorithms*, volume 2. MIT press Cambridge.
- Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429–1450.
- Delgado, A., Jensen, R. M., Janstrup, K., Rose, T. H., and Andersen, K. H. (2012). A constraint programming model for fast optimal stowage of container vessel bays. *European Journal of Operational Research*, 220(1):251–261.
- Desrosiers, J. and Lübbecke, M. E. (2005). *A primer in column generation*. Springer.
- Erera, A. L., Morales, J., and Savelsbergh, M. (2005). Global intermodal tank container management for the chemical industry. *Transportation Research Part E*, 41(6):551–566.
- Gendreau, M. and Potvin, J.-Y. (2010). *Handbook of metaheuristics*, volume 2. Springer.
- Gendron, B., Crainic, T. G., and Frangioni, A. (1999). *Multicommodity capacitated network design*. Springer.
- Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138.
- Kim, H.-J., Chang, Y.-T., Kim, K.-T., and Kim, H.-J. (2012). An epsilon-optimal algorithm considering greenhouse gas emissions for the management of a ship’s bunker fuel. *Transportation Research Part D: Transport and Environment*, 17:97 – 103.
- Meng, Q. and Wang, S. (2011). Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):695–708.
- Mulder, J., Dekker, R., and Sharifyazdi, M. (2012). Designing robust liner shipping schedules: Optimizing recovery actions and buffer times. *Report / Econometric Institute, Erasmus University Rotterdam*.
- Pacino, D. (2012). Fast generation of container vessel stowage plans. *Ph. D. Thesis, IT University of Copenhagen*.

- Pacino, D., Delgado, A., Jensen, R., and Bebbington, T. (2011). Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. *Computational Logistics*, 6971:286–301.
- Pacino, D. and Jensen, R. (2012). Constraint-based local search for container stowage slot planning. *Lecture Notes in Engineering and Computer Science*, 2:1467–1472.
- Pacino, D. and Jensen, R. (2013). Fast slot planning using constraint-based local search. *IAENG Transactions on Engineering Technologies*, 186:49–63.
- Plum, C., Pisinger, D., and Jensen, P. (2015). Bunker purchasing in liner shipping. In *Handbook of Ocean Container Transport Logistics*, volume XVII, pages 251–278. Springer.
- Plum, C., Pisinger, D., and Sigurd, M. M. (2014). A service flow model for the liner shipping network design problem. *European Journal of Operational Research*, 235(2):378–386.
- Plum, C. E. M. and Jensen, P. N. (2007). Minimization of bunker costs. *Master thesis, University of Copenhagen*.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374.
- Rossi, F., Van Beek, P., and Walsh, T. (2006). *Handbook of constraint programming*. Elsevier.
- Sheng, X., Lee, L., and Chew, E. (2014). Dynamic determination of vessel speed and selection of bunkering ports for liner shipping under stochastic environment. *OR Spectrum*, 36(2):455–480.
- Shintani, K., Imai, A., Nishimura, E., and Papadimitriou, S. (2007). The container shipping network design problem with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 43(1):39–59.
- Song, D.-P. and Dong, J. X. (2012). Cargo routing and empty container repositioning in multiple shipping service routes. *Transportation Research Part B: Methodological*, 46(10):1556–1575.
- Tierney, K., Áskelsdóttir, B., Jensen, R. M., and Pisinger, D. (2013). Solving the liner shipping fleet repositioning problem with cargo flows. *Transportation Science*, Articles in Advance.

- Toth, P. and Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications*. SIAM.
- Vielma, J. P. (2015). Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57.
- Wang, S. and Meng, Q. (2015). Robust bunker management for liner shipping networks. *European Journal of Operational Research*, 243(3):789–797.
- Yao, Z., Ng, S. H., and Lee, L. H. (2012). A study on bunker fuel management for the shipping liner services. *Computers and Operations Research*, 39:1160 – 1172.

## Part II

# Network Optimization with Transit Time Restrictions



# The Time Constrained Multi-commodity Flow Problem

with D. Pisinger, S. Røpke and B.D. Brouer<sup>1</sup>

---

## Abstract

The multi-commodity network flow problem is an important sub-problem in several heuristics and exact methods for designing route networks for container ships. The sub-problem decides how cargoes should be transported through the network provided by shipping routes. This paper studies the multi-commodity network flow problem with transit time constraints which puts limits on the duration of the transit of the commodities through the network. It is shown that for the particular application it does not increase the solution time to include the transit time constraints and that including the transit time is essential to offer customers a competitive product.

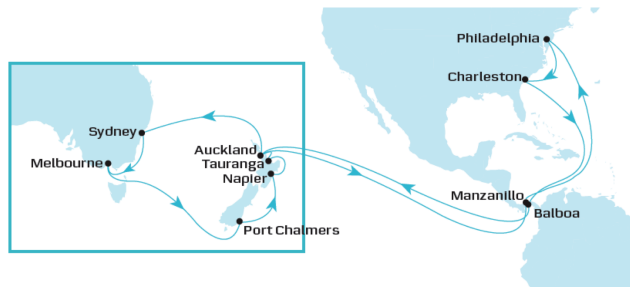
---

<sup>1</sup>Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76, 122-138.

### 3.1 Introduction

According to IMO (2014) 90% of global trade is carried out via the sea, and ships flying EU flags emit more than 20 million tons of  $CO_2$  (MaritimeCO2, 2014). Container Shipping involves the transportation of a major share of the worlds goods and has been steadily growing (with a small decrease around 2009 due to the economic crisis). Reliance on container shipping to transport goods internationally is only expected to increase due to its economic advantages compared to other transportation modes. Additionally, the  $CO_2$  emissions per ton cargo transported using maritime transport is significantly lower than road and rail transport. Hence, even small improvements in the underlying network of a liner shipping company can have a significant impact, both economically and environmentally. Despite this, the *Liner Shipping Network Design* (LSND) problem has not received a lot of attention in the Operations Research literature and it is far from being a well-solved problem (Meng et al., 2014). Christiansen et al. (2004) and Christiansen et al. (2013) provide comprehensive reviews of the literature published within the field of maritime optimization and liner shipping.

A liner shipping network consists of a number of rotations, which are round trips. It is common to have weekly departures at each port, hence a sufficient number of vessels are deployed to each rotation, to ensure the requested frequency. Figure 3.1 shows an example of a real-world rotation. Different vessels have varying capacity and speed, and the transport of a commodity through the network may include the use of several rotations to connect between the origin and destination port. The switch from one rotation to another is referred to as *transshipment* and there is a cost associated with this since the container must be handled by the quay-cranes at the transshipment ports and



**Figure 3.1:** An example of a sailing route (rotation) in the Maersk Line network. *Source: Maersk (2014).*

possibly stored temporarily at the container yard. On top of this, the container will experience a wait time during the transfer process. Because of the associated cost and transit time and the risk of goods being damaged containers are at most subject to a few transshipments, when traveling from their origin to destination. The *transit time* is the time it takes a commodity to travel from origin to destination. Transit time is counted in days and allowed transit times may vary from one day to several months Brouer et al. (2014). Liner shipping networks that are optimized only with respect to cost get an unrealistically high network utilization as containers are allowed on detours that offer unused capacity but in practice they will violate transit time restrictions. Given a candidate network a multi-commodity network flow (MCF) problem is solved in order to decide, which of the available cargoes should be shipped on which routes. An extensive treatment of the MCF problem can be found in e.g., Ahuja et al. (1993). The MCF problem can be formulated as a linear programming problem which can be solved in polynomial time and there are many algorithms for solving it. One of these methods is by delayed column generation, see Desaulniers et al. (2005) and Ahuja et al. (1993). In order to include the transit time constraint one has to solve an extended version of the problem, the time constrained MCF problem, which is NP-hard. This is easily shown by reduction from the shortest weight constrained path problem, (Garey and Johnson, 1979). (We transform this problem into a time constrained MCF by having only one commodity with source and destination as given by the shortest path problem.) This paper presents an algorithm for the time constrained MCF problem and given the LSND application several possible improvements are presented.

To the best of our knowledge, most algorithms for the LSND problem do not include transit time restrictions for shipped commodities. This paper studies the consequence of neglecting the transit time restrictions in existing networks. This is done by taking the networks produced by a LSND heuristic and comparing the estimated revenue with and without including the time constraint in the cargo flow calculations. The results show a substantial difference, and we therefore recommend that future LSND algorithms should include the transit time constraint if possible.

The remainder of the paper is organized as follows. In Section 3.2 we discuss the *level of service* in liner shipping and review relevant literature. In Section 3.3 we introduce the multi-commodity flow problem with time constraints and describe a delayed column generation procedure for solving it. Furthermore, we discuss a way to tailor the resource constrained shortest path problem, which arise as the sub-problem in the column generation process, in order to solve it efficiently. Section 3.5 describes a contraction scheme for the graph, which



reduces the number of edges in certain instances of the graph to speed up the sub-problem computations. Section 3.6 introduces novel ways of modeling the transshipments to accommodate different network design model scopes. Finally, we conduct computational experiments in Section 3.7 and investigate the sensitivity of the travel time restrictions.

## 3.2 The Level of Service in Liner Shipping

Several factors such as price, transit time, transshipments, port coverage, frequency, reliability, administration, equipment, environmental friendliness and schedules can be relevant and important for a shipper when considering different carriers, (Brouer et al., 2014). Hence it is important to meet these constraints when constructing and evaluating liner shipping networks. The cost and transit time are often identified as the most important factors, (Meng et al., 2014; Brouer et al., 2014; Gelareh et al., 2010; Notteboom and Vernimmen, 2009; Notteboom, 2006), however, most previous work within LSND neglects transit time and mainly considers cost.

Designing networks with focus only on cost has the apparently attractive benefit that reducing cost goes hand in hand with reducing  $CO_2$  emissions as fuel is the largest cost component. Reducing  $CO_2$  emissions is an important goal of several governments, and it is generally attractive for carriers as well as shippers to have a green profile. Slow steaming is one common way of both reducing cost and emissions, but this requires a broader introduction of the level of service requirements in the network design models. There is an inherent trade off between reducing bunker consumption and thereby emissions through speed reduction and offering competitive transit times for commodities.

On the other hand, by offering a time competitive mode of transport more cargo will be transported this way, reducing the global  $CO_2$  emissions. By introducing a maximum transit time for each commodity in the network, the number of allowed paths will be limited significantly for the individual commodities and introduces new limits on the feasible solutions in the network design process. However, it requires adding a time dimension to all edges in a network and especially the service time at ports and the time spent transshipping between rotations need careful analysis to obtain both competitive network cost and transit times. In order for a network to be competitive it must offer low transit times and few transshipments.

Implications of travel time restrictions is not well-studied in connection with

LSND, but recently it has been studied in connection with related problems. Agarwal and Ergun (2008) present a time-space graph to introduce a rough schedule of weekdays in the network design process, but they do not introduce travel time restrictions and do not account for the cost of transshipping goods. Gelareh et al. (2010) study a hub-and-spoke network design problem for two liner shipping companies in a competitive environment. The market share is determined by transit time and transportation cost. Wang and Meng (2011) study schedule design and container-routing for a given network with predefined paths. They minimize the transshipment cost, add a penalty cost for longer transit times and a bonus for shorter transit times. Wang and Meng (2012) give a tactical model for schedule design, where they minimize the cost, while maintaining a required transit time taking time uncertainty into account. Meng and Wang (2012) study the fleet deployment problem in conjunction with transit time levels in a space-time network. Wang et al. (2013) study an integer program for generating a container path for a single OD-pair taking transit time and cabotage rules into account. A case study considering a single path is presented. No computational run time is reported. Plum et al. (2014) consider transit time for the design of a single rotation with up to 25 ports. Finally, Wang and Meng (2014) present a non-linear mixed integer model for the network design problem taking transit time into account and formulate a column generation based heuristic for solving it for a Europe Asia network with 12 ports. Álvarez (2011) gives mathematical expressions for the transit time of goods, which is composed of time at sea, time at ports and dwell time, and derive a bi-linear cost expression for the inventory holding.

Neither exact nor heuristic solution methods are yet able to solve LSND instances with the size of a global carrier to (near) optimality, but a promising approach is to rely on a two-tier structure as in Álvarez (2011); Brouer and Desaulniers (2012); Brouer et al. (2014), where route planning, fleet deployment and sailing speed is determined in the upper tier corresponding to determining the cost of the network, while the lower tier determines the revenue of the network by flowing the available cargo. In the following, we consider the cargo flow subproblem, which is one of the main challenges in LSND. In Brouer et al. (2011) a specialized MCF considering liner shipping cargo flow with empty repositioning is presented along with a computational study of solving the LP arc-flow model versus solving a path-flow model using column generation. Holmberg and Yuan (2003) discuss general MCF problems with side constraints and propose a column generation procedure for solving them, the solution method in this paper is similar to that of Holmberg and Yuan (2003), but specialized to the LSND application.

It is worth mentioning that graph representations and commodity flows within

the maritime area are studied outside the core Operations Research community. Examples are Kaluza et al. (2010), Ducruet and Notteboom (2012) and Ducruet (2013) who create aggregate graphs representing vessel movements by combining the historic trajectories of individual vessels. The papers analyse the aggregate graphs, for example with respect to change over time (Ducruet and Notteboom, 2012) or with respect to the importance of diversification of port activities (Ducruet, 2013).

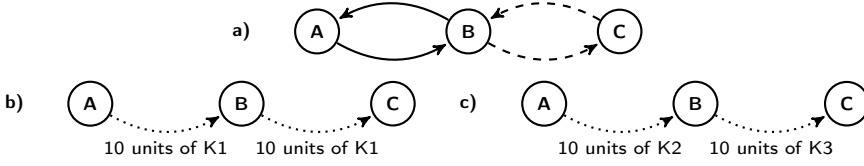
### 3.3 Time-constrained Multi-commodity Flows

As mentioned above, a promising approach for solving the LSND problem heuristically is to use a two phase approach. The first phase builds a network consisting of a number of rotations and the second phase decides, how cargo should be transported in this network to evaluate the cost/revenue of the network. In this paper we do not consider network design, instead we focus solely on determining how cargo should flow through the network.

Figure 3.2a) illustrates a basic network that is the output of Phase 1. In this example the network is composed of two rotations  $R1$  and  $R2$ . In general the graph contains the node set  $N$  as consisting of  $P$  and  $C$ , ports and calls respectively. Goods can be transshipped between rotations at the port, where rotations meet. I.e. goods can be transshipped between rotation  $R1$  and  $R2$  in node  $B$ . The flow of goods through the network is decided in the second phase. This is illustrated in Figures 3.2b) and 3.2c). In this example we only have three commodities. Ten units of commodity  $K1$  is based in node  $A$  and destined for node  $C$ , ten units of  $K2$  is based in  $A$  and destined for  $B$ , and ten units of  $K3$  is based in  $B$  and destined for  $C$ . Transporting one unit of commodity  $K1$ ,  $K2$ , and  $K3$  results in an income of 10, 4, and 4, respectively. The capacities of the edges in the network is determined by the capacities of the vessel class used and the frequency of the rotation. In this example we assume that all the edges have a capacity of 10. When solving the cargo flow problem no cost is associated with traversing the voyage edges, as we assume the sailing cost to be roughly identical whether or not the ship is fully loaded, an assumption that is not completely true in practice <sup>2</sup>. The cost of operating the ships will be accounted for in Phase 1. In the cargo flow phase we do pay for each transshipment action and loading and unloading. In this example, we assume, that the cost is one per unit transshipped and neglect load/unload costs.

---

<sup>2</sup>In reality a typical container ship uses more fuel when traveling fully loaded compared to sailing empty.



**Figure 3.2:** a) A simple example network with two rotations  $R1$  (solid edges) and  $R2$  (dashed edges). b) and c) show two possible flows in the network. The paths for commodities  $K1$ ,  $K2$ , and  $K3$  are marked by dotted edges.

Figure 3.2b) shows the optimal solution to the cargo flow problem in our example. We can only transport a total of 10 units through the two rotations  $R1$  and  $R2$  and hence, transporting 10 units of commodity  $K1$  gives the highest revenue (90). Now consider that the traversal of each edge and each transshipment action takes one time unit and consider that all commodities must reach their destination within 2 time units. Therefore, the solution found without transit time restrictions, is no longer feasible. The optimal solution given the time restriction is shown in Figure 3.2c). Here it is possible to ship commodity  $K2$  and  $K3$ . The resulting revenue is 80 since we do not have to pay for the transshipment operation.

In the following we first review the MCF problem and later show how a time-constrained MCF can be modeled and solved.

The arc flow formulation MCF problem can be stated as follows. We redefine  $G = (N, A)$  to be a generic, directed graph with nodes  $N$  and edges  $A$ . Let  $K$  be the set of commodities to transport and  $b^k$  be the amount of commodity  $k \in K$  that is available for transport. We assume that each commodity has a single origin node and a single destination node denoted  $o(k)$  and  $d(k)$ , respectively. Let  $u_{ij}$  be the capacity of edge  $(i, j)$ . For each node  $i \in N$  and commodity  $k \in K$  we define

$$b(i, k) = \begin{cases} b^k & \text{if } i = o(k) \\ -b^k & \text{if } i = d(k) \\ 0 & \text{otherwise} \end{cases}$$

and for each node  $i \in N$  we define the sets  $\delta^+(i) = \{(j, j') \in A : j = i\}$  and  $\delta^-(i) = \{(j, j') \in A : j' = i\}$ , that is, the set of edges with tail and head in node  $i$ , respectively. The model uses decision variables  $x_{ij}^k$  that specify the amount of commodity  $k \in K$  that flows through edge  $(i, j)$ . We do not impose integrality conditions on the flow as in practice several thousand containers are moved on a single vessel and hence fractional containers are negligible. Additionally the demand is often a forecast so the variation in this will exceed rounding errors.

Brouer et al. (2011) investigate the effect of integrality for the version of the problem without time-constraints and find that most solutions are integral in practice and that the gap in terms of objective value for the considered real-life instances never exceeds 0.01% if a fractional solution is just rounded. For each unit of commodity  $k$  that flows through edge  $(i, j)$  the cost is  $c_{ij}^k$ . With this notation the MCF problem can be stated as a linear programming problem as follows:

$$\min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k \quad (3.1)$$

subject to

$$\sum_{(j,j') \in \delta^+(i)} x_{jj'}^k - \sum_{(j,j') \in \delta^-(i)} x_{jj'}^k = b(i, k) \quad i \in N, k \in K \quad (3.2)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} \quad (i, j) \in A \quad (3.3)$$

$$x_{ij}^k \geq 0 \quad (i, j) \in A, k \in K \quad (3.4)$$

The objective function (3.1) minimizes the cost of the chosen flow, constraint (3.2) ensures flow conservation and ensures that commodities originates and terminates in the right nodes. Constraint (3.3) ensures that the capacity of each edge is respected. This formulation has  $|K||A|$  variables and  $|A| + |K||N|$  constraints. The number of variables is hence polynomially bounded, but for large graphs like the ones seen in global liner shipping networks this formulation requires excessive computation time and may even be too large for standard linear programming solvers (see e.g. Brouer et al. (2011)).

It is not hard to see how the MCF can be used to find the optimal cargo flow in the LSND given a set of rotations, but we would like to make a comment on transshipments and rejected demands. The most straightforward approach for modeling transshipments is to model each transshipment port by a node for each rotation that visits the port. Edges between nodes from different rotations, meeting at a transshipment ports, are used to model the actual transshipment. The cost of such edges is equal to the cost of the transshipment. Section 3.6 discusses this and other modeling approaches. The standard MCF model written above enforces that all demands are being met. We can let the model reject demand by including dummy arcs between source and destination with an appropriate penalty.

An alternative model for the MCF is the path-flow formulation where each variable corresponds to a path through the graph for a certain commodity. To

define the model we need to define the following sets: let  $\Omega^k$  be the set of all feasible paths for commodity  $k$ ,  $\Omega^k(a)$  be the set of paths for commodity  $k$  that uses edge  $a$  and  $\Omega(a) = \cup_{k \in K} \Omega^k(a)$  is the set of all paths that use edge  $a$ . We have a variable  $x_j$  for each path  $j$ . The variable states, how many units of a specific commodity that is routed through the given path, the cost of each variable is given by the parameter  $c_j$ . The model is:

$$\min \quad \sum_{k \in K} \sum_{j \in \Omega^k} c_j x_j \quad (3.5)$$

$$s.t. \quad \sum_{j \in \Omega^k} x_j = b^k \quad k \in K \quad (3.6)$$

$$\sum_{j \in \Omega(a)} x_j \leq u_{ij} \quad (i, j) \in A \quad (3.7)$$

$$x_j \geq 0 \quad k \in K, j \in \Omega^k \quad (3.8)$$

Here constraint (3.6) ensures that the demand of each commodity is met and constraint (3.7) ensures that the capacity limit of each edge is obeyed. The path-flow model has  $|A|+|K|$  constraints, but the number of variables is, in general, growing exponentially with the size of the graph. However, using delayed column generation the necessary variables can be generated dynamically and in practice the path-flow model can often be solved faster than the arc-flow model for large scale instances of the LSND problem (see Brouer et al. (2011)).

Delayed column generation works with a reduced version of the LP (3.5)-(3.8), which is called the master problem. The master problem is defined by a reduced set of columns  $\bar{\Omega}^k$  for each commodity  $k$  such that a feasible solution to the LP (3.5)-(3.8) can be found using variables from  $\cup_{k \in K} \bar{\Omega}^k$  (if there is no available connection a forfeited edge with a penalty cost is used.). Solving this LP gives rise to dual variables  $\pi_k$  and  $\lambda_{ij}$  corresponding to constraint (3.6) and (3.7), respectively. For a variable  $j \in \cup_{k \in K} \Omega^k$  we let  $\kappa(j)$  denote the commodity that a variable serves and let  $p(j)$  represent the path corresponding to the variable  $j$ , represented as the set of edges traversed by the path. Then we can calculate the *reduced cost*  $\bar{c}_j$  of each variable  $j \in \cup_{k \in K} \Omega^k$  as follows:

$$\bar{c}_j = \sum_{(i,j) \in p(j)} (c_{ij}^{\kappa(j)} - \lambda_{ij}) - \pi_{\kappa(j)}.$$

If we can find a variable  $j \in \cup_{k \in K} (\Omega^k \setminus \bar{\Omega}^k)$  such that  $\bar{c}_j < 0$  then this variable has the potential to improve the current LP solution and should be added to the master problem, which is resolved to give new dual values. If, on the other hand, we have that  $\bar{c}_j \geq 0$  for all  $j \in \cup_{k \in K} (\Omega^k \setminus \bar{\Omega}^k)$  then we know the master problem defined by  $\bar{\Omega}^k$  provides the optimal solution to the complete

problem (for more details see Álvarez (2009); Ahuja et al. (1993)). In order to find a variable with negative reduced cost or prove that no such variable exists we solve a sub-problem for each commodity. The sub-problem seeks the feasible path for commodity  $k$  with minimum reduced cost given the current dual values. It is not hard to see that solving this problem amounts to solving a shortest path problem from source to destination of the commodity with edge costs given by  $c_{ij} - \lambda_{ij}$  and subtracting  $\pi_k$  from this cost in order to get the reduced cost. We note that  $\lambda_{ij} \leq 0$ , which means that the edge cost in the sub-problem will be non-negative.

We add a constraint on the transit time of the voyage of each commodity to accommodate the transit time restrictions. Adding this constraint to the arc-flow model is non-trivial since the demand for each commodity can be fulfilled using multiple paths. In this formulation multiple paths are bundled up in a tree structure, where the time of each individual path cannot easily be tracked. In the path-flow formulation the constraint can be handled in the definition of  $\Omega^k$ , ensuring that the set only contains paths that are feasible with respect to the transit time constraint. The formulation separates each of the paths into a single variable, enabling us to track time of each individual commodity. However, doing so complicates the delayed column generation algorithm since the sub-problem has to ensure that the transit time of each generated path is less than or equal to the maximum transit time for the given commodity. This changes in this case the sub-problem from being an ordinary shortest path problem solved e.g. using Dijkstra's algorithm to a weakly NP-hard *resource constrained shortest path*, RCSP, problem (Hassin, 1992).

### 3.3.1 Detailed Network Description

We define the set of voyage edges,  $A_v$ , as the set of edges connecting two nodes in  $C$  on the same rotation, i.e. consecutive port calls on a rotation and  $A_v = \{(i, j) | i, j \in C \wedge i, j \in r'\}$ . The time,  $t_a$ , to traverse arc  $a \in A_v$  is calculated according to the distance sailed with the average speed of the rotation. An edge connecting two calls in the same port is denoted a transshipment edge belonging to the edge set  $A_t = \{(i, j) | i, j \in C \wedge i \in r_1, j \in r_2\}$ .  $t_a$  for  $a \in A_t$  denotes the transshipment time and  $c_a$  for  $a \in A_t$  the transshipment cost. As we do not have a schedule in the following we work with an average transshipment time of three days, i.e.,  $t_a = 3$  for  $a \in A_t$ . Every load and unload of a unit of cargo is associated with a cargo handling cost. Hence, for the set of (un)load edges,  $A_l = \{(i, j) | (i \in C \wedge j \in P) \vee (j \in C \wedge i \in P)\}$ ,  $t_a$  for  $a \in A_l$  denotes the handling time and  $c_a$  for  $a \in A_l$  denotes the load/unload cost. We set the load and unload time to one day, i.e.,  $t_a = 1 \forall a \in A_l$ . Lastly, it is possible

to omit a cargo using the set of forfeited edges  $A_f = \{(i, j) | (i, j \in P) \wedge (\exists k \in K, o(k) = i \wedge d(k) = j)\}$ .  $t_a$ , for  $a \in A_f$  denotes the maximum allowed transit time and  $c_a$  is a goodwill penalty for not transporting the cargo. We assume that the loading and unloading as well as transshipment times in a port are independent of the number of containers to be handled at the port. The edge set  $A$  is defined as  $A = A_v \cup A_t \cup A_l \cup A_f$ .

### 3.4 Resource Constrained Shortest Path Calculations

The RCSP sub-problem can be solved using various methods. One method is to use a label setting algorithm as proposed in Irnich and Desaulniers (2005). Labeling algorithms are based on dynamic programming and use resource extension functions and dominance functions to efficiently calculate the shortest path through a graph considering several resources, here (reduced) cost and time. The resources must be of a form where they can be determined at the vertices of a directed walk in a graph. We say that a resource is *constrained* if there is at least one vertex in the graph where the resource is bounded from above, otherwise the resource is *unconstrained*. We treat (reduced) cost as an unconstrained resource, which we minimize, and time as a constrained resource, as the limits on transit time, limits the time resource in the algorithm. When solving the MCF problem using the labeling algorithm the accumulated consumption of the resources is non-decreasing in each extension of a label. This is a prerequisite for the algorithm to work. Labels are used to store the information on the resource values for (incomplete) paths through the graph. Labels are associated with the vertices in the graph and they are propagated via resource extension functions along the edges in the graph. An extension of a label is feasible if the resulting label is feasible, i.e. the transit time did not exceed the limit. A decisive feature of the algorithm is to keep the number of labels as small as possible. This is done via a dominance function, which eliminates unnecessary labels. The dominance function checks if all resources, i.e. cost and time, for one label is less than or equal to the value of the resource in the other label at each vertex, i.e., a label,  $l_a$ , dominates another label,  $l_b$ , if  $cost(l_a) \leq cost(l_b)$  and  $time(l_a) \leq time(l_b)$ . This improves the running time of the algorithm, since dominated labels need not to be extended and can be deleted. Pseudo code is given in Figure 3.3.

At each iteration, the labeling algorithm selects a label from the set of unprocessed labels  $U$  and checks it for dominance and feasibility. If the label is



**Require:** a graph,  $G$ , with corresponding node and edge descriptors  
**Require:** a node descriptor,  $s(k)$ , for the start node of a path  
**Require:** a set of node descriptors,  $E$ , containing destinations of demands with origin  $s(k)$   
**Initialize** Initialize the set of unprocessed label  $U = \{s\}$   
 $T = \max(\text{allowed transit time of all commodities leaving } s)$   
**while**  $U \neq \emptyset$  **do**  $\text{current\_label} \leftarrow \min(U)$   
    **if**  $\text{current\_label}$  is not dominated **then**  
        node  $i = \text{ResidentNode}(\text{current\_label})$   
        check dominance and delete dominated and processed labels  
        mark  $\text{current\_label}$  as processed  
        **for all** outgoing edges,  $(i, j)$ , of  $i$  **do**  
             $\text{new\_label} = \text{resource\_extension\_function}(\text{current\_label})$   
            **if**  $\text{new\_label.time} > T$  (i.e. not feasible) **then** delete  $\text{new\_label}$   
            **else**  $U \leftarrow \text{new\_label}$   
        **else** delete  $\text{current\_label}$   
**for all**  $e \in E$  **do** add paths,  $pe$ , and resource consumption for  $e(k)$  to  $PE$   
    **for all**  $pe \in PE$  **do**  
        **if** transit time violate allowance for  $pe$  **then** delete path  
        **else** add path to set of feasible paths,  $FE$

**Figure 3.3:** Pseudo code for the resource constrained shortest path algorithm; o-all implementation.

dominated it is deleted, whereas if it is undominated, it is extended along all out-edges of the current vertex. If the new label is also feasible it is added to the set of unprocessed labels and to the set of labels residing at the successor vertex. If the new label is not feasible, it is deleted. The algorithm stops, when there are no more unprocessed labels. Then it determines whether the destination vertex can be reached and constructs all undominated (Pareto-optimal) paths. Hence, tight limits on transit time in a large network will cause the algorithm to terminate faster as fewer labels need to be extended.

### 3.4.1 Reducing the Number of RCSP Calculations

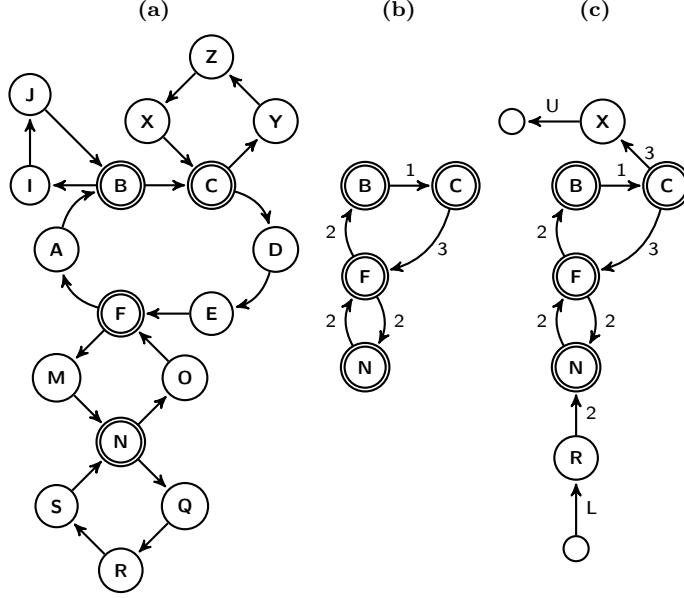
In the column generation procedure of the MCF problem a RCSP problem must be solved for each of the commodities with individual restrictions on travel time for all commodities. However, the natural origin-destination (o-d) implementation for each commodity suggested by the MCF problem can be modified. The RCSP algorithm is executed for the commodity with the maximum allowed

transit time from the set of commodities with identical origin. As a “by product” the shortest paths for the remaining commodities with identical origin are also found. This is due to the nature of the label setting algorithm, where labels represent paths. All labels that are not dominated (reduced cost and time) and do not violate the travel time restriction for the commodity with the longest allowed are not deleted. Hence, we are guaranteed to find all optimal paths for the commodities with origin  $o$  if such exist. At the end of the algorithm all paths to a node are considered and a post processing procedure that erases paths violating the allowed transit time for each commodity is implemented, see the pseudo code in Figure 3.3. Hence, at most the number of ports  $|P|$  RCSP calculations are needed to obtain o-d paths for all commodities, but still it is possible to use the domination. If using the o-d implementation of the algorithm we would need  $|K|$  calculations. For a global network the number of ports is significantly less than the number of commodities  $|P| < |K|$ . In the WorldSmall instance provided in Brouer et al. (2014) there are  $|P| = 47$  ports and  $|K| = 1764$  commodities and in the AsiaEurope instance there are  $|P| = 111$  ports and  $|K| = 4000$  commodities.

The algorithm is based on a variation of the Boost Graph Library (BGL) implementation. It uses a resource extension function to specify extensions of labels, and a dominance function comparing cost and time for two labels.

## 3.5 Graph Contraction

The computational time increases with the size of the graph, but due to the inherent structure of the networks in Liner Shipping it is possible to simplify the corresponding graphs for each of the sub-problems. Figure 3.4a) shows a graph representation of the voyage edges in a small instance with five rotations ( $B \rightarrow I \rightarrow J$ ), ( $C \rightarrow Y \rightarrow Z \rightarrow X$ ), etc. All edges have a cost of one. There are four minor hubs  $B$ ,  $C$ ,  $F$ , and  $N$ , where transshipments from one rotation to another are possible. We contract this graph to one where only hub nodes are kept and edges represent voyage possibilities between hubs. This graph is shown in Figure 3.4b). An edge in this graph is a contraction of one or more edges from the original graph. The edge  $C \rightarrow F$  for example represents the path  $C \rightarrow D \rightarrow E \rightarrow F$  in the original graph. The simplified graph does not contain all nodes from the original graph so many of the needed shortest path computations are not possible in the reduced graph. However, for each necessary shortest path computation we extend the graph as necessary. This is illustrated in Figure 3.4c). The figure shows the graph that is necessary to compute the shortest path from  $R$  to  $X$ . Nodes  $R$  and  $X$  are added to the graph. Node  $R$  connects to the contracted network through node  $N$  so an edge



**Figure 3.4:** A graph representation of the voyage edges in a small instance with five rotations. All edges have a cost of one. There are four minor hubs  $B$ ,  $C$ ,  $F$ , and  $N$ , where transshipments from one rotation to another are possible. Load, unload, and transshipment edges have been excluded for simplicity.

is added from  $R$  to  $N$  with the appropriate cost (the cost of  $R \rightarrow S$  plus the cost of  $S \rightarrow N$ ) and node  $X$  can only be reached through node  $C$  so an edge is added from  $C$  to  $X$  with appropriate cost. Also a load edge,  $L$ , is included to account for loading cost and time as well as an unload edge,  $U$ . Hence in contrast to Brouer and Desaulniers (2012) the graph only includes relevant load/unload edges. Figure 3.5 shows the pseudo code for the contraction algorithm. After the contraction of the graph, it is modified separately for each commodity or commodity group such that edges connecting the load port and the destination port(s) with the contracted network are added if these are not hubs. Likewise, load and unload edges are added.

As mentioned in Section 3.4.1 we prefer to do shortest path calculations with a single origin and many destinations. The multi-destination calculations are also possible in the contracted graphs by adding appropriate edges for each destination in the same way as described for a single destination shortest path calculation. The *Reduced Graph* decreases the number of extensions needed in

**Require:** a graph,  $G$ , with edges and nodes corresponding to the transportation network and a copy,  $G'$ , only containing the nodes

**for all** rotations,  $r$ , in  $G$  **do**

find degree of nodes in  $r$  to determine whether it is a transshipment node.

**if** # transshipment nodes  $> 1$  **then** determine *first\_voyage\_edge* on  $r$

**while** next node  $\neq$  first node **do** find next port and voyage edge on  $r$

add current voyage edge info to update *current\_contracted\_edge*

**if** *degree\_destination\_node*  $\leq 2$  **then** continue

**else** add *current\_contracted\_edge* to  $G'$

clear *current\_contracted\_edge*

**Figure 3.5:** Pseudo-code for contracting a graph.

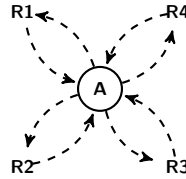
the label setting algorithm for the shortest path calculations and hence speed up the computation. This approach is more tractable when only a few ports are hubs (i.e. visited by more than one rotation). We use the network structure presented in Brouer and Desaulniers (2012) as a reference, denoted the *Full Graph*. Additionally to reduce the size of the reduced graph used during the SPP calculations we only consider the load and unload edges which are relevant to the considered set of commodities as well as the relevant forfeited edges in contrast to Brouer and Desaulniers (2012).

## 3.6 Representation of Transshipments

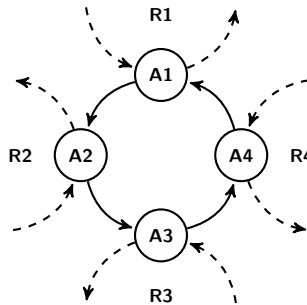
There are several ways of handling transshipments in the graph. Each modeling approach has different properties and benefits. An alternative modeling approach to the ones presented in the following is given in Plum et al. (2013).

Figure 3.6-3.11 show different graph representations of a transshipment structure. In most cases a port node is augmented to contain internal port nodes and edges such that the cost, capacity, and time of the port operation can be correctly accounted for. We are going to analyse the structures in Figure 3.8-3.10 in further detail in the computational section while we just want to mention some additional properties of the structures shown in Figure 3.6-3.11.

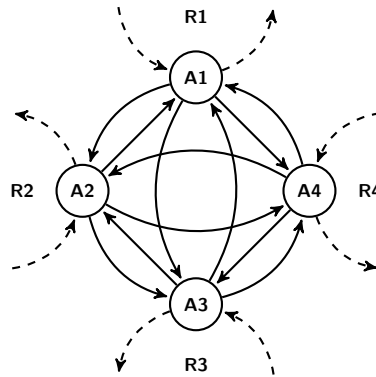
The structure in Figure 3.6 is the most basic representation of a transshipment and it does not allow modeling of neither cost nor time related to transshipments as commodities transfer directly between the rotations. No additional nodes or edges are added and a commodity will transfer directly from one



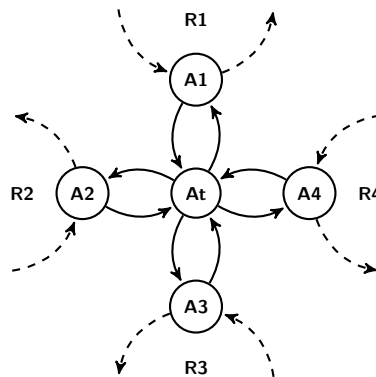
**Figure 3.6:** Simple transshipment structure shown for a physical port,  $A$ , with four rotations visiting the port. Voyage edges are dashed and cargo transship directly from one voyage edge to another.



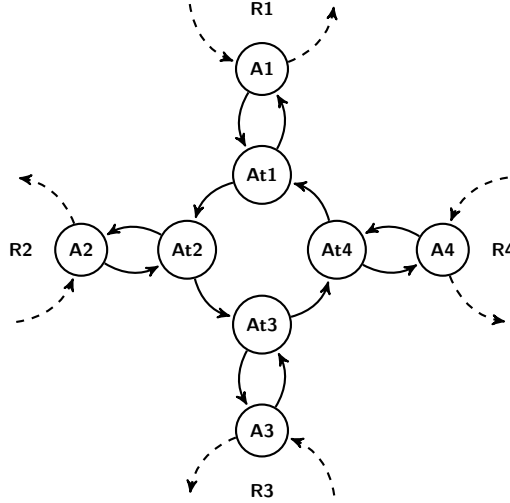
**Figure 3.7:** The transshipment structure shown for a physical port,  $A$ , with four rotations visiting the port.  $A1$ ,  $A2$ ,  $A3$ , and  $A4$  are the corresponding port calls and they are connected in a ring. Solid edges correspond to transshipment edges and dashed to voyage edges.



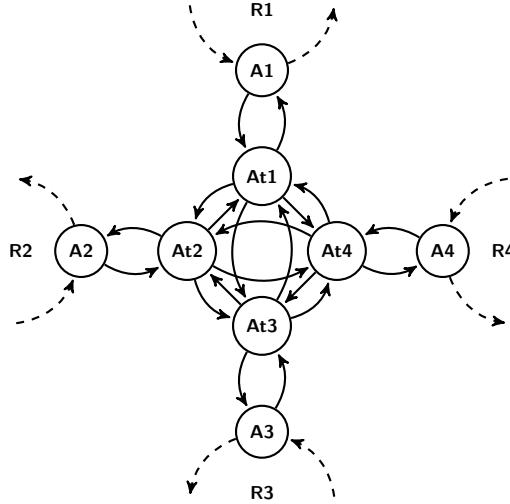
**Figure 3.8:** The *complete* transshipment structure shown for a physical port,  $A$ , with four rotations visiting the port.  $A1$ ,  $A2$ ,  $A3$ , and  $A4$  are the corresponding port calls. Solid edges correspond to transshipment edges and dashed to voyage edges for the four different rotations visiting port  $A$ .



**Figure 3.9:** The *star* transshipment structure shown for a physical port,  $A$ , with four rotations visiting the port.  $A1$ ,  $A2$ ,  $A3$ , and  $A4$  are the corresponding port calls and  $At$  is an extra transshipment node. Solid edges correspond to transshipment edges and dashed to voyage edges for the four different rotations visiting port  $A$ .



**Figure 3.10:** The *ring* transshipment structure shown for a physical port,  $A$ , with four rotations visiting the port.  $A1$ ,  $A2$ ,  $A3$ , and  $A4$  are the corresponding port calls.  $At1$ ,  $At2$ ,  $At3$ , and  $At4$  are extra transshipment nodes. Solid edges correspond to transshipment edges and dashed to voyage edges for the four different rotations visiting port  $A$ .



**Figure 3.11:** A general transshipment structure shown for a physical port,  $A$ , with four rotations visiting the port.  $A1$ ,  $A2$ ,  $A3$ , and  $A4$  are the corresponding port calls.  $At1$ ,  $At2$ ,  $At3$ , and  $At4$  are extra transshipment nodes and edges are added between all pairs of these. Solid edges correspond to transshipment edges and dashed to voyage edges.

voyage edge to another. Figure 3.7 shows a generalized version of this simple structure where each port call is assigned a transshipment node and these are connected in a “ring”. This requires  $r$  extra edges and  $r$  extra nodes, where  $r$  is the number of rotations visiting a hub. This allows modeling of a schedule and the time between two services (if ordered in terms of arrival) can be added to the edges. It is however not possible to correctly account for transshipment costs and buffer time.

The complete structure found in Figure 3.8 is used in e.g. Brouer and Desaulniers (2012), and may be the most intuitive representation of a transshipment as all rotations visiting a port are directly connected to all other rotations visiting the same port. This allows different costs and transit times between different rotations, which can be calculated directly according to some given schedule including buffer time. This comes at a cost of having a high number of edges in larger hubs. It requires  $r(r - 1)$  edges and  $r$  nodes. Each edge has an associated cost and time. The representations in Figure 3.9 and in Figure 3.10, denoted *star* and *ring* respectively, mitigate these costs by introducing one or several additional transshipment nodes. The number of edges only increases linearly with the number of rotations visiting a hub.

The ring structure, like the complete structure, allows individual transit times based on a given schedule, whereas the star structure does not. The star structure, which is also used in Wang and Meng (2013), introduces one new transshipment node,  $At$ , and has  $2r$  edges and  $r + 1$  nodes. All rotations are connected to the transshipment node via an edge with an associated cost and time. Edges out of the transshipment node have no associated cost or time. The ring structure introduces a new transshipment node for each port, in the figure  $At1$ ,  $At2$ ,  $At3$ , and  $At4$  respectively, and new edges in a “ring” with an associated time and cost. The edges leaving the transshipment nodes has no associated cost or time in our experiments but as discussed below adding cost or time allows modeling of additional properties. The structure has  $3r$  edges and  $2r$  nodes. Both the complete and ring structure makes it possible to consider an actual schedule with arrival and departure time specified, however it is not possible to take buffer time between two rotations into account in the ring structure. The star structure offers a simpler structure than the complete and ring structure if average transit times are considered and not actual schedules. Both the ring structure and the star structure can additionally handle operational capacities in the port such as quay crane capacity, i.e. adding a capacity to the edges between  $A$  and  $At$  will ensure that the number of containers loaded and unloaded to/from all services in the port does not exceed that capacity of the quay or the cranes assigned to a given vessel. This is not possible to model in the complete structure. In the cases, where the total travel time is close



to the limit, it is sufficient to check the initial transshipment edge to cut off all possible transfers in the star and ring structure. Hence for the purpose of evaluating the algorithmic effects of considering cargo transit times we use the structures in Figure 3.8 - 3.10 in the computational experiments.

Finally, the structure in Figure 3.11 is a generalization of the discussed models where it is possible to take both port productivity in terms of crane and quay capacity as well as buffer time between two rotations into account. This structure requires  $r(r - 1) + 2r$  edges and  $2r$  nodes.

In practice, to reduce the number of edges further, we can combine the structures such that for all physical ports if the number of visiting rotations  $> 3$  (i.e. it is a hub with more than 3 visiting rotations) we change transshipment layout to either the star or ring structure, whereas for hubs with  $\leq 3$  rotations visiting we use the complete structure in all cases.

## 3.7 Computational Experiments

The algorithms are implemented in C++ and run on a normal laptop with an Intel Core i5 2.60GHz and 16 GB Ram using one core. We use the Boost Graph library to handle the networks and solve the LPs using the COIN-OR solver. We investigate the influence of the transit time limits, the graph contraction and different transshipment structures as well as sensitivity in the following. The results can be seen in Table 3.4 - 3.8 and Figure 3.12-3.15.

### 3.7.1 Data

The data instances used are based on the benchmark instances in *Liner-lib* (Brouer et al., 2012) published along with Brouer et al. (2014). We use networks constructed based on six of these instances, see Table 3.1. The networks have been constructed using the matheuristic that does not consider transit time described in Brouer and Desaulniers (2012). We report results for networks from each instance. These networks are denoted *Baltic (Bal)*, *West Africa (WAF)*, *Mediterranean (Med)* *Pacific (Pac)*, *WorldSmall (WS0)*, *AsiaEurope (AE0)*. Furthermore, we consider additional large networks of varying quality, denoted *WorldSmall1 (WS1)*, *WorldSmall2 (WS2)*, *WorldSmall3*

Instance	Ports	Demands
<b>Single hub instances</b>		
Bal	12	22
WAF	19	38
<b>Multi hub instance</b>		
Med	39	369
<b>Trade lane instances</b>		
Pac	45	722
AE	111	4,000
<b>World instance</b>		
WS	47	1,764

**Table 3.1:** The instances considered. Consult Brouer et al. (2014) for further details.

# transshipment edges	Transshipment structure		
	Complete	Star	Ring
Bal ( 13 voyage & 26 load edges)	22	12	17
WAF (43 voyage & 86 load edges)	166	64	98
Med (64 voyage & 128 load edges)	90	78	108
Pac (153 voyage & 306 load edges)	734	278	410
WS0 (275 voyage & 550 load edges)	2,076	534	797
AE0 (308 voyage & 616 load edges)	1,530	526	773

**Table 3.2:** Number of transshipment edges for the different structures for the two largest instances. The first column gives the number of transshipment edges for the complete transshipment structure, the second column correspond to the star structure, and the third column to the ring structure.

(*WS3*), *AsiaEurope1* (*AE1*), *AsiaEurope2* (*AE2*), *AsiaEurope3* (*AE3*). Table 3.2 shows the number of transshipment edges for the considered instances. For the instance AE0 the network consist of 308 voyage edges, 1,530 transshipment edges, 616 load/unload edges, and 4,000 forfeited edges and 111 ports and 308 rotation vertices corresponding to port calls, i.e., a total of 2,454 edges (6,454 including the forfeited edges) and 422 nodes. WS0 correspondingly has 2,901 edges (4,665 including the forfeited edges) and 322 nodes. Edge costs are calculated as described in Brouer et al. (2014) using the data given in *Liner-lib*.

Table 3.3 shows the number of voyage edges in the graph for different instances. For different commodities we get slightly different graphs and hence the number of edges varies for the commodities by a few edges. The first column is

the average number of contracted edges in the reduced graph for the o-d implementation of the RCSP algorithm. The second column states the average number of contracted voyage edges when using the o-all implementation of the RCSP algorithm, while the last column gives the number of voyage edges in the *full* graph used in Brouer and Desaulniers (2012).

# voyage edges	Reduced (o-d)	Reduced (o-all)	Full
Bal	7	7	26
WAF	36	37	43
Med	49	51	64
Pac	146	148	153
WS0	271	274	275
AE0	280	287	308

**Table 3.3:** The average number of voyage edges in the reduced graph for an o-d and o-all representation compared to the number of voyage edges in the full graph. Dijkstra and o-all RCSP uses the o-all representation for the reduced graph, whereas the o-d RCSP uses the o-d representation for the reduced graph. The full graph is the same for all algorithms.

As seen in Table 3.2 and 3.3, it has a significant effect to contract edges in smaller instances, where the networks are less complex and only few of the ports are visited by several rotations. However, for larger networks there are only very few edges that can be contracted because the majority of the ports serve several rotations. In all instances the number of load and unload edges is reduced as discussed earlier. If it was possible to identify ports where transshipments are not allowed or possible it would be possible to omit these nodes and reduce the graph further. In the next sections we consider the effects of time limits, the implementation of the shortest path algorithm, the transshipment structure, the graph reduction and finally the sensitivity of the time limits. For all instances we report the computational run times in seconds to solve the full MCF problem to optimality, i.e., no more columns with reduced cost are found for any commodity.

### 3.7.2 Effect of Transit Time Limits and Implementation

Imposing realistic limits on the transit time for the individual commodities actually has a significant positive effect on the computational tractability. Even though it requires the solution of a more complex RCSP evaluation as subproblem, the vastly reduced solution space yields faster computations in almost all instances than when using Dijkstra’s algorithm for the unconstrained

problem. See Table 3.4 and Table 3.5 for a comparison of the instances. It is clear that the RSCP is only faster when implemented to take advantage of the problem structure.

Table 3.6 compares the implementation of RCSP as an *origin-destination* (o-d) implementation where the MCF problem is solved for all origin-destination pairs and an *origin-all* (o-all) implementation where all commodities with same origin is considered in one iteration of the RCSP-algorithm. In both cases we solve the problem to optimality considering all demands. Clearly the o-all implementation is advantageous with speed-ups in all larger instances and all discussed transshipment structures up to a factor of 9. The average speed-up for the considered instances using the full graph is 7 and 4 for the reduced graph. For the setting used in (Brouer and Desaulniers, 2012) with the complete transshipment structure and the full graph, the average speed-up is 2 when using the o-all RCSP compared to Dijkstra’s algorithm, see Table 3.4 and Table 3.5 for a comparison of the instances. For the o-d implementation, the vast majority of the time is spent solving the sub-problem. For the o-all implementation for most instances more than half of the time is spent adjusting and solving the LP. Looking at Table 3.6 the o-all implementation with limits on transit time yields on average a speed-up of 1.5 for the considered larger instances on the reduced graph compared to the full graph.

Table 3.4 shows that the RSCP is only faster to solve when tight time limits are indeed imposed. The left part of the table shows run times and number of column generation iterations for instances where time limits are not imposed and the right part of the graph shows run times and number of column generation iterations with the limits imposed. On average for the considered instances of different size and structure the speed-up is 4 with the maximum being 9 when the time limits are imposed. Table 3.7 reveals that the share of containers shipped drops dramatically when transit times are imposed on networks designed without considering these. For several of the instances the utilization drops more than 30 percent point.

### 3.7.3 Effect of Transshipment Structure

The effect of the different transshipment structures can be studied from Tables 3.6 and 3.4. Comparing the star transshipment structure to the complete structure reveals an average speed-up of less than 1.5. The ring structure only gives slight speed-up. However, the speed-up is more significant, when comparing the results for the reduced graph and the full graph adopted from (Brouer et al., 2014).

o-all	No transit time limits					With transit time limits					Itr
	Time (s)	It	Time (s)	It	Vol	Time (s)	It	Time (s)	It	Vol	
	Star		Complete		(%)	Star		Complete		(%)	
	Full Graph										
Bal	0.002/0.001	20	0.002/0.001	292	1	0.001/0.001	20	0.001/0.001	292	1	19
WAF	0.017/0.012	60	0.023/0.016	694	9	0.005/0.003	30	0.007/0.004	365	0	19
Med	0.213/0.095	60	0.214/0.099	695	3	0.065/0.037	30	0.066/0.039	360	5	36
Pac	1.97/0.722	12	2.14/0.905	129	1	0.262/0.170	30	0.339/0.232	351	5	45
WS0	15.7/3.99	17	16.8/5.19	169	1	2.69/1.54	9	3.85/2.50	967	4	47
AE0	63.6/16.3	15	69.1/19.8	169	1	19.0/8.96	11	20.8/10.9	127	5	111
	Reduced Graph										
Bal	0.001/0.001	20	0.002/0.001	292	1	0.001/0.001	20	0.001/0.001	292	1	19
WAF	0.012/0.005	60	0.017/0.009	694	9	0.005/0.002	30	0.006/0.003	365	0	19
Med	0.151/0.023	70	0.151/0.025	795	3	0.047/0.014	40	0.048/0.016	460	5	36
Pac	1.44/0.203	13	1.58/0.343	129	1	0.154/0.059	30	0.227/0.114	351	5	45
WS0	13.0/0.99	17	14.1/2.16	169	1	1.79/0.56	9	2.82/1.34	967	4	47
AE0	48.2/1.82	15	47.9/3.14	179	1	12.0/1.80	11	13.2/2.92	117	5	111

**Table 3.4:** Comparison of run times (overall solution time / time spent in sub-problem) when imposing limits on travel times for the full and reduced (contracted) graph with the star and complete transshipment structure. The RCSP o-all algorithm is used. *It* indicates the number of column generation iterations to reach optimality. *Vol (%)* gives the volumes of cargo shipped. *Itr* gives the number of times the RCSP algorithm is called in each column generation iteration.

Dijkstra, full graph	Time (s) Star	Time (s) Complete	Vol (%)
Bal	0.001	0.001	92.1
WAF	0.006	0.006	94.9
Med	0.115	0.102	95.3
Pac	1.23	1.36	91.1
WS0	11.0	13.9	91.3
AE0	46.0	42.0	91.2

**Table 3.5:** Benchmark results for Dijkstra’s algorithm, as in (Brouer and Desaulniers, 2012), for instances varying in size with no limits on travel time.

	Time (s) Complete	Time (s) Star	Time (s) Ring	Time (s) Complete	Time (s) Star	Time (s) Ring
	RCSP o-d			RCSP o-all		
	Full Graph					
AE1	132 / 118	104 / 90	128 / 109	20 / 8	18 / 6	24 / 7
AE2	100 / 90	82 / 74	100 / 87	13 / 6	13 / 5	16 / 6
AE3	118 / 108	95 / 83	116 / 103	16 / 7	14 / 6	19 / 7
WS1	21 / 20	13 / 12	16 / 14	3 / 2	2 / 1	3 / 1
WS2	26 / 25	13 / 13	15 / 14	3 / 2	2 / 1	2 / 1
WS3	29 / 28	16 / 15	20 / 17	3 / 2	2 / 1	3 / 1
	Reduced Graph					
AE1	63 / 50	37 / 23	44 / 26	15 / 3	13 / 2	18 / 2
AE2	46 / 38	27 / 18	37 / 25	10 / 2	9 / 1	12 / 2
AE3	48 / 39	27 / 18	39 / 24	11 / 2	10 / 1	14 / 2
WS1	10 / 9	4 / 3	6 / 4	2 / 1	1 / 0.3	2 / 0.4
WS2	13 / 11	4 / 3	7 / 5	2 / 1	1 / 0.4	2 / 1
WS3	16 / 14	6 / 4	8 / 6	2 / 1	2 / 0.4	2 / 1

**Table 3.6:** Comparison of the different transshipment structures, graph constructions and algorithms. The upper part of the table shows runtimes (overall solution time to reach optimality for the column generation procedure / time spent in sub-problem) for the full graph and the lower part shows results for the reduced (contracted) graph. Complete, star, and ring refer to the different transshipment structures.

### 3.7.4 Effect of Reducing the Graph

The effect of contracting and reducing the number of edges in the graph can be observed from Tables 3.4 and 3.6. The main reduction comes from the reduced solution time of the sub-problems. The effect is clearly more pronounced for the larger instances, WS and AE, with an average overall speed-up above a factor of 2. The maximum speed-up is 3, while the average speed-up for the mid-size instances is only 1.3. The speed-up gained from reducing the graph is both a product of contracting the edges and removing the forfeited commodity edges included in (Brouer et al., 2014).

In general there are no unambiguous conclusion regarding the number of iterations in the column generation, but the column generation takes longer time for networks with a high percentage of volume shipped and for larger networks. Additionally, disregarding the forfeited commodity edges used in (Brouer and Desaulniers, 2012) in the RCSP implementation is important and for smaller

	Vol (%)	
	With transit time limits	No transit time limits
AE1	76.0	92.1
AE2	70.0	95.3
AE3	77.0	91.1
WS1	57.6	94.9
WS2	54.8	91.3
WS3	61.9	91.2

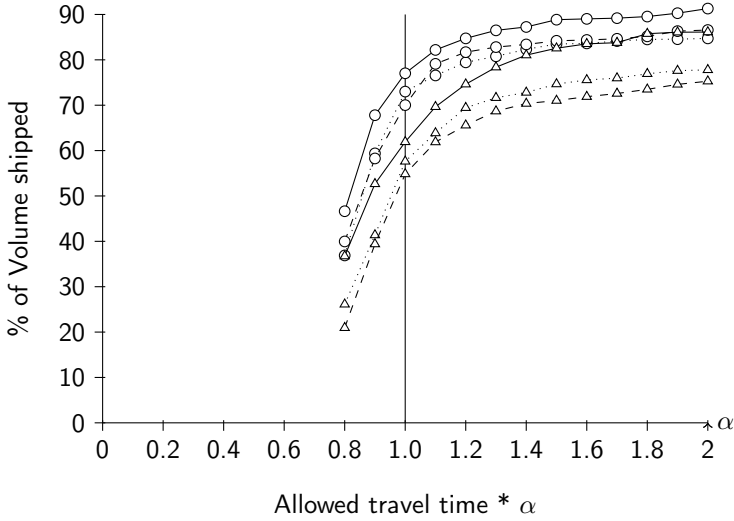
**Table 3.7:** The volumes shipped in the large instances when considering transit time limits compared to the volumes when transit time limits are not imposed.

instances the graph contraction makes the problem easier to solve. However, keeping the forfeited edges for the commodities in question aids bounding the algorithm.

### 3.7.5 Sensitivity of Travel Time Restrictions

Clearly the limits on travel time affect the size of the solution space and hence the amount of cargo that can flow through the network. Figure 3.12 shows how varying the limits between 80% and 200% of the limits given in *Liner-lib*, affects the amount of cargo with feasible paths in the 6 large instances described in Table 3.1. It is important to notice that the network optimization is done without considering travel times. At the default allowed travel time ( $\alpha = 1.0$ ) as little as 51% of the goods can be transported in one instance, while up to around 90% of the goods can be transported when doubling the allowed travel times, see Table 3.8 or Figure 3.12.

In an optimized network of the type WS (WS0) where around 91 % of the goods can be transported when limits on travel times are not considered, the implications of time constraints is investigated further. Figure 3.13 shows that a slight increase of the volume of goods that can be transported can be obtained through the network when adjusting the time limits by a factor of 2.5 to 10. Figure 3.14 shows the revenue that can be obtained through the network, when adjusting the limits by a factor of 0.9 to 2.0. At the default allowed travel time ( $\alpha = 1.0$ ) only around 60 % of the maximum revenue (obtained without restrictions on travel time) can be obtained, while around 99 % of the unconstrained revenue for this network can be obtained when doubling the



**Figure 3.12:** Sensitivity of limits on travel time based on the data given in (Brouer et al., 2014). The three upper instances (circles) are AE1 (dotted), AE2 (dashed), and AE3 (solid). The three lower instances (triangles) are WS1 (dotted), WS2 (dashed), and WS3 (solid).

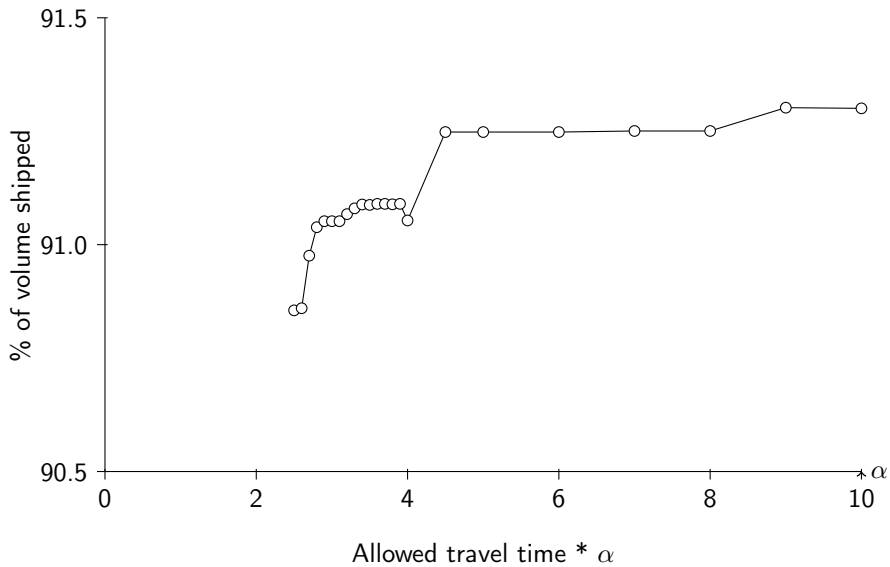
allowed travel times. Figure 3.15 shows that a slight increase in revenue can be obtained, when adjusting the limits by a factor of 2.5 to 10. This clearly shows that some commodities will take very long undesirable paths through the network if there are no limits on travel time. A summary for the instances of different size can be seen in Table 3.8.

Note that the percentage of volume shipped as a function of allowed travel time is not necessarily a monotonically increasing function, whereas the revenue as a function of allowed travel time is. This is because more profitable cargo can become available on alternative paths as the time limits are increased. Furthermore, note that the effect is more significant for larger instances than smaller instances.

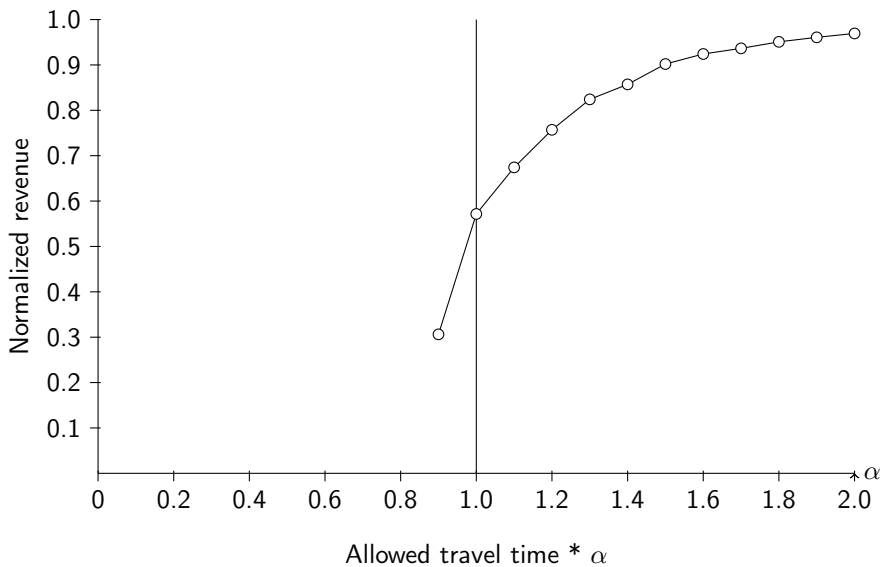
## 3.8 Conclusions

The presented analysis clearly shows that it is relevant and necessary to consider limits on travel times in the network design process. Omitting the transit

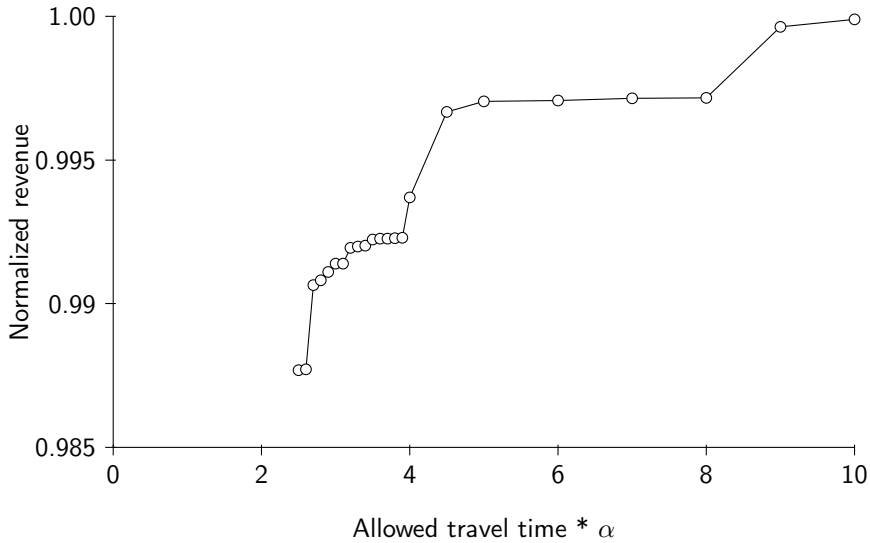




**Figure 3.13:** Sensitivity of limits on travel time based on the data for an instance of WS (WS0) given in (Brouer et al., 2014).



**Figure 3.14:** Sensitivity of limits on travel time based on the data for an instance of WS (WS0) given in (Brouer et al., 2014).



**Figure 3.15:** Sensitivity of limits on travel time for an instance of WS (WS0) based on the data in (Brouer et al., 2014).

Instance	1 · time	2 · time	20 · time
Baltic (vol.)	1 (92%)	1 (92%)	1 (92%)
Baltic (rev.)	1	1	1
WAF (vol.)	0.67 (65%)	1.00 (95%)	1 (95%)
WAF (rev.)	0.42	0.99	1
MED (vol.)	0.63 (61%)	0.90 (86%)	1 (95%)
MED (rev.)	-0.76	0.51	1
Pacific (vol.)	0.57 (51%)	0.97 (88%)	1 (91%)
Pacific (rev.)	-0.30	0.92	1
WS (vol.)	0.74 (67%)	0.98 (89%)	1 (91%)
WS (rev.)	0.57	0.97	1
AE (vol.)	0.83 (75%)	0.96 (88%)	1 (91%)
AE (rev.)	0.76	0.96	1

**Table 3.8:** Normalized volumes transported (vol.) and normalized revenues (rev.) for the different instances under different transit time limits and compared to no transit time limits imposed. In column two, three, and four, numbers in parenthesis indicate the absolute amount of goods transported. The first column shows the implication of the actual transit time limit, while we allow twice the time in column two and 20 times the allowed time in column three. 20 times the allowed time, in practice corresponds to no restrictions on travel time.

time constraint when designing routes lead to cargo being transported along intricate routes that would not be accepted in practice. It could be feared that including transit time constraints in the MCF problem would lead to much higher computational times, but the present experiments show that this is not the case for the instances under study. The proposed graph contractions and simpler transshipment structures further help speeding up the solution time of the time constrained multi-commodity network flow problem. The obvious next step is to include the proposed algorithm for the time constrained multi-commodity network flow problem into heuristics for solving the liner shipping network design problem.

One should also take into account, that when designing a liner shipping network, the actual departure times (the schedule) are not fixed yet, meaning that transshipment times are only estimates. Hence, instead of using a very tight constraint on the transshipment time, a soft punishment could be used for exceeding the maximum allowed transshipment time up to a given upper limit. This could e.g. be a quadratic punishment also giving a reward for transshipment times below the limit. The punishment somehow indicates how difficult it will be to subsequently design a schedule that meets the time constraints. This is all easily handled by having a complete list of transit times and costs from solving the shortest path problem using a dynamic programming algorithm.

## Acknowledgements

The authors wish to thank Christian Plum at Maersk Line and three anonymous referees for valuable comments. This project was supported in part by The Danish Strategical Research Council and The Danish Energy Technology Development and Demonstration Program (EUDP) under the ENERPLAN and GREENSHIP projects and in part by The Danish Maritime Fund under the Competitive Liner Shipping Network Design project.

## Bibliography

- Agarwal, R. and Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196.
- Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Prentice hall.
- Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11:186–208.
- Álvarez, J. F. (2011). Mathematical expressions for the transit time of merchandise through a liner shipping network. *Journal of the Operational Research Society*, 63(6):709–714.
- Brouer, B., Álvarez, J., Plum, C., Pisinger, D., and Sigurd, M. (2012). Liner-lib 2012. [www.linerlib.org](http://www.linerlib.org).
- Brouer, B., Álvarez, J., Plum, C., Pisinger, D., and Sigurd, M. (2014). A base integer programming model and benchmark suite for liner shipping network design. *Transportation Science*, 48(2):281–312.
- Brouer, B. and Desaulniers, G. (2012). A matheuristic for the liner shipping network design problem. In *Liner Service Network Design (PhD thesis)*, chapter 5. Technical University of Denmark, Department of Management Engineering.
- Brouer, B., Pisinger, D., and Spoorendonk, S. (2011). The cargo allocation problem with empty repositioning (caper). *INFOR*, 49(2):109–124.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Christiansen, M., Fagerholt, K., and Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18.
- Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2005). *Column generation*, volume 5. Springer.
- Ducruet, C. (2013). Network diversity and maritime flows. *Journal of Transport Geography*, 30:77–88.
- Ducruet, C. and Notteboom, T. (2012). The worldwide maritime network of container shipping: spatial structure and regional dynamics. *Global Networks*, 12(3):395–423.

- Garey, M. R. and Johnson, D. S. (1979). Computers and intractability: a guide to the theory of np-completeness. *WH Freeman & Co., San Francisco*.
- Gelareh, S., Nickel, S., and Pisinger, D. (2010). Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):991–1004.
- Hassin, R. (1992). Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42.
- Holmberg, K. and Yuan, D. (2003). A multicommodity network-flow problem with side constraints on paths solved by column generation. *INFORMS Journal on Computing*, 15(1):42–57.
- IMO (2014). International maritime organization (imo).
- Irnich, S. and Desaulniers, G. (2005). *Shortest path problems with resource constraints*. Springer.
- Kaluza, P., Kölzsch, A., Gastner, M. T., and Blasius, B. (2010). The complex network of global cargo ship movements. *Journal of the Royal Society Interface*, 7(48):1093–1103.
- Maersk (2014). The maersk line service network.
- MaritimeCO2 (2014). Impact assessment for the adoption of co2 emission trading for maritime transport.
- Meng, Q. and Wang, S. (2012). Liner ship fleet deployment with week-dependent container shipment demand. *European Journal of Operational Research*, 222(2):241–252.
- Meng, Q., Wang, S., Andersson, H., and Thun, K. (2014). Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280.
- Notteboom, T. E. (2006). The time factor in liner shipping services. *Maritime Economics & Logistics*, 8(1):19–39.
- Notteboom, T. E. and Vernimmen, B. (2009). The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337.
- Plum, C. E., Pisinger, D., Salazar-González, J.-J., and Sigurd, M. M. (2014). Single liner shipping service design. *Computers & Operations Research*, 45:1–6.

- Plum, C. E., Pisinger, D., and Sigurd, M. M. (2013). A service flow model for the liner shipping network design problem. *European Journal of Operational Research*.
- Wang, S. and Meng, Q. (2011). Schedule design and container routing in liner shipping. *Transportation Research Record: Journal of the Transportation Research Board*, 2222(1):25–33.
- Wang, S. and Meng, Q. (2012). Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B: Methodological*, 46(5):615–633.
- Wang, S. and Meng, Q. (2013). Reversing port rotation directions in a container liner shipping network. *Transportation Research Part B: Methodological*, 50:61–73.
- Wang, S. and Meng, Q. (2014). Liner shipping network design with deadlines. *Computers & Operations Research*, 41:140–149.
- Wang, S., Meng, Q., and Sun, Z. (2013). Container routing in liner shipping. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):1–7.



# Time Constrained Liner Shipping Network Design

with B.D. Brouer, G. Desaulniers, and D. Pisinger<sup>1</sup>

---

## Abstract

We present a mathematical model and a solution method for the liner shipping network design problem. The model takes into account coordination between vessels and transit time restrictions on the cargo flow. The solution method is an improvement heuristic, where an integer program is solved iteratively to perform moves in a large neighborhood search. To assess the effects of insertions/removals of port calls, flow and revenue changes are estimated for relevant commodities along with an estimation of the change in the vessel cost. Our improvement heuristic is applicable as a real-time decision support tool for a liner shipping company. It can be used to find improvements to the network when evaluating changes in operating conditions or testing different scenarios. Computational results on the benchmark suite *Liner-lib* are reported. For these instances we obtain insight about the optimal operational speed of the vessels, as well as statistics on which cargo reaches the destination within the requested time limit. Furthermore, we demonstrate that the algorithm can construct high-quality networks for most instances. Finally, we show how the algorithm can be extended to introduce a limit on the number of transshipments for each container.

---

<sup>1</sup>Karsten, C. V., Brouer, B. D., Desaulniers, G., and Pisinger, D. (2015). Time Constrained Liner Shipping Network Design. Submitted to: Transportation Research Part E special issue on “Computational Logistics at Work: Coordination and Control in Transport Logistics.” **and** Brouer, B. D., Desaulniers, G., Karsten, C. V., and Pisinger, D. (2015). A matheuristic for the liner shipping network design problem with transit time restrictions. *Computational Logistics*, volume 9335 of *Lecture Notes in Computer Science*, pages 195–208. Springer International Publishing.



## 4.1 Introduction

The *time constrained liner shipping network design problem*, TCLSNDP, is a core planning problem faced by container carriers. The problem is to design a set of cyclic routes, *services*, for container vessels to provide transport for goods while respecting cargo travel time restrictions. The objective of the problem is to maximize the profit of the liner shipping company through the revenues gained from container transport taking into account the fixed cost of deploying vessels and the variable cost related to the operation of the routes and the handling cost of cargo transport. As a consequence of maximizing profits the liner shipping network design problem generally allows rejection of some commodities if deemed unprofitable.

Liner shipping companies offer a range of services that are operated according to a published schedule with a fixed frequency to make it easier for customers to plan ahead. *Scheduling* decisions refer to the temporal aspect of the vessel routings and include the timing of events along the entire round trip. A *fleet* of vessels is deployed to the services such that the capacity and speed is in accordance with the demand maintaining the desired frequency. Global carriers generally deploy vessels with similar characteristics to a service to reduce the complexity of the network design and corresponding schedules (Notteboom and Vernimmen, 2009; Stopford, 2009). The services give rise to a network of related ports. Containers, or more generally commodities, are transported through the network from port A to port B, and a transport may include the use of several services to connect between the origin and destination ports. The transits between services are referred to as *transshipments* and the *transit time* is the time used to transport a container from origin to destination. Each transport must respect a maximal transit time restriction of each individual commodity. In practice transit times vary from one day to several months and most containers are transshipped no more than twice corresponding to a feeder - main line - feeder connection. However, some containers can be subject to up to five or six transshipments increasing the risk of delays and total handling time. Generally, customers prefer transports with no or few transshipments. The number of transshipments can vary based on the origin and destination regions. A network with direct connections between all serviced ports would offer low transit times and no transshipments, but at the same time it would be very expensive to operate for the carrier as most pairs of ports do not have enough container demand to fill a vessel. This illustrates the trade-off faced by liner shipping companies between the cost of networks versus transit time and/or number of transshipments offered to the customers. Providing low freight rates by minimizing the cost of the network is likely to result in prolonged transit times as exemplified in Karsten et al. (2015). Likewise designing a network to

minimize transit times is likely to result in a very costly network since speed increases.

The main costs associated with the fleet include fuel cost, port and canal charges, and financing of vessels (this includes capital costs of acquiring or financing a vessel and the operational cost (OPEX) which includes crew, maintenance, and insurance). Stopford (2009) estimated the fuel cost to be 35-50 % of a vessel's operational cost, capital cost to be 30-45 %, OPEX to be 6-17 %, and port cost to be 9-14 %. This obviously depends on the fuel price and general economic environment. The cargo handling cost is calculated from the load and unload cost at the origin and destination ports and the cost associated with transshipments at intermediate ports. In addition to this, there are costs for the customer associated with owning or leasing containers. The load and unload costs do not depend on the routing of the container, whereas the transshipment cost does. Revenues are obtained by transporting cargo through the network and varies based on the type of cargo and the level of service offered.

Recent literature on the liner shipping network design problem, LSNBP, allows arbitrary transit times for all commodities (Brouer et al., 2014b; Liu et al., 2014; Wang and Meng, 2014; Mulder and Dekker, 2014; Brouer et al., 2014a; Plum et al., 2014; Reinhardt and Pisinger, 2012; Gelareh et al., 2010; Agarwal and Ergun, 2008) although it is generally acknowledged that transit times are decisive for the competitiveness of the network design, e.g. Brouer et al. (2014a). Initial work to construct a multi-criteria objective function is presented in Álvarez (2012) that considers a bi-linear expression for the inventory cost of the cargo on board vessels, but the level of service calculations are not computationally tractable in the already very complex liner shipping network design models. However, the inventory cost of commodities on board vessels is only indirectly a concern to the carrier, when excessive transit times result in the customers switching to a different carrier. Hence, the carriers concern is to ensure a maximal transit time corresponding to the market level of service. Wang and Meng (2014) introduce deadlines on commodities in a non-linear, non-convex mixed-integer programming (MIP) formulation of liner shipping network design with transit time restrictions. As a consequence the model does not allow transshipment of cargo, which is another common trait of the liner shipping network design problem.

Brouer et al. (2014b) develop a matheuristic for the LSNBP. The matheuristic is an *improvement* heuristic according to the categorization in the survey on matheuristics by Archetti and Speranza (2014) meaning that an integer program is used as a move operator. The present paper extends the method of Brouer et al. (2014b) to include transit times. Álvarez (2012) presents math-

emathical expressions for the inventory cost of containers during transport. No computational results are reported as the mathematical expressions are not easily incorporated into existing models of the LSNDP. In Wang et al. (2013) an integer program for deciding minimum cost container paths for a single OD pair respecting transit time and cabotage restrictions is considered. Karsten et al. (2015) present a column generation algorithm for a time constrained multicommodity flow (MCF) problem applied to a liner shipping network. A resource constrained shortest path problem is solved for each origin using a specialized label setting algorithm. Different topologies of graphs for liner shipping networks are presented. Computational results for solving the MCF problem with and without transit times on global-sized liner shipping networks are reported. The solution times for the time constrained MCF problem is comparable to solving the MCF problem without transit time restrictions. The algorithm of Karsten et al. (2015) is used in the matheuristic presented in this paper for evaluating a given network during the search. A liner shipping network design problem considering transit time restrictions is presented in Wang and Meng (2014). The model excludes transshipments between services. The problem is proven to be NP-hard and is formulated as a non-linear, non-convex mixed integer program. A column-generation-based heuristic is developed and a case study is presented for a network of 12 main ports on the Asia-Europe trade lane with three different vessel classes. The model is suggested as an aid to planners in a liner shipping company and the case study provides high-quality network suggestions and important insights to assist the planners. The authors suggest incorporation of transshipments along with transit time restrictions as an area of future research.

Meng et al. (2014) and Christiansen et al. (2013) provide broader reviews of recent research on routing and scheduling problems within liner shipping.

In this paper we present a capacitated multicommodity network design formulation for the TCLSNDP allowing for an arbitrary number of transshipments and enabling restrictions on transit time of individual commodities. We propose an adaptation of the matheuristic of Brouer et al. (2014b) to show that it is possible to incorporate the transit time restrictions in a heuristic context. Furthermore, we show that it is tractable to incorporate a limit on the number of transshipments for each commodity. The benchmark instances presented in Brouer et al. (2014a), *Liner-lib*, are used for the computational results of this paper. The benchmark instances contain maximum transit time for all OD pairs.

The rest of this paper is organized as follows: Section 4.2 introduces our mathematical model. Section 4.3 extends the IP used as a move operator in Brouer

et al. (2014b) to also consider transit times. Section 4.4 reports computational results for the benchmark instances. Section 4.5 shows the sensitivity to the provided transit times and Section 4.6 extends the algorithm and results with a limit on the number of transshipments. We end the paper by drawing conclusions and discussing extensions in Section 4.7.

## 4.2 Mathematical Model for the TCLSNBP

In the following we introduce the notation used to formulate the TCLSNBP mathematically. An instance of the TCLSNBP consists of the following sets:

- $P$ : Set of ports with an associated port call cost  $c_p^e$ , unit (un)load cost  $c_U^p, c_L^p$ , unit transshipment cost  $c_T^p$  and berthing time  $b_p$  spent on a port call.
- $K$ : Set of demands, where each demand has an origin  $O_k \in P$ , a destination  $D_k \in P$ , a quantity,  $q_k$ , a revenue per unit,  $z_k$ , a reject penalty per unit  $\tilde{z}_k$  and a maximal transit time,  $t_k$ .
- $E$ : Set of vessel classes with specifications for the weekly charter rate,  $f_e$ , capacity  $U_e$ , minimum ( $v_{min}^e$ ) and maximum ( $v_{max}^e$ ) speed limits in knots per hour, fuel consumption  $g_v^e$  as a function of the speed, and fuel consumption  $h^e$  per hour, when the vessel is idle at ports. There are  $N_e$  vessels available of class  $e \in E$ . The price for one metric ton of fuel is denoted  $c_B$ .
- $D$ : Matrix of the direct distances  $d_{ij}^e$  between all pairs of ports  $i, j \in P$  and for all vessel classes  $e \in E$ . The distance may depend on the vessel class draft as the Panama canal is draft restricted. Along with  $d_{ij}^e$  follows an indication of the cost  $l_{ij}^e$  associated with a possible traversal of a canal.

A solution to the TCLSNBP is a subset of the set of feasible services  $S$ . A service consists of a set of ports  $P' \subseteq P$ , a number of vessels, and an average sailing speed. A service is cyclic but may be non-simple, that is, ports can be visited more than once. In this model we allow a single port to be visited twice, yielding a so-called *butterfly* route. A weekly frequency of port calls is obtained by deploying multiple vessels to a service. Let  $e(s) \in E$  be the vessel class assigned to a service  $s$  and  $n_{e(s)}$  the number of vessels of class  $e(s)$  required to maintain a weekly frequency. A round trip may last several weeks but due

to the weekly frequency exactly one round trip is performed every week. Let  $v_s$  be the service speed in nautical miles per hour.

The mathematical model of the TCLSNDP relies on a set of service variables and a path flow formulation of the underlying time constrained MCF problem. To describe the service network of the TCLSNDP, we define  $F^s$  to be the port sequence  $p_1^s, p_2^s, \dots, p_m^s$  for the service  $s \in S$ . Let  $|s|$  denote the number of unique ports in a service  $s \in S$  and  $|F^s| = m$  the number of port calls in  $s$ .

Furthermore we define a directed graph,  $G(V, A)$ , with vertices  $V$  and arcs  $A$ .  $V = V_P \cup V_R$  is the set of vertices, where  $V_P$  is the subset of vertices representing the unique ports  $p \in P$ , and  $V_R$  is the subset of service vertices representing all port calls by all services.  $V_R = \bigcup_{s \in S} V_{F^s}$  and  $V_{F^s}$  is the subset of vertices representing the port calls  $p_1^s, p_2^s, \dots, p_m^s$  of service  $F^s$ ,  $s \in S$ .  $p(v)$  is a function mapping a vertex  $v \in V_R$  (i.e., a port call) to its actual port  $p \in P$ . The set of arcs in the graph can be divided into (un)load arcs, transshipment arcs, sailing arcs, and forfeited arcs, i.e.  $A = A_L \cup A_U \cup A_T \cup A_S \cup A_K$ . These sets are formally defined below and we associate with each arc  $a \in A$  a cost  $c_a$ , traversal time  $t_a$ , and capacity  $C_a$ .

- $A_L = \{(p, v) \mid p \in V_P, v \in V_{F^s}\}$  and  $A_U = \{(v, p) \mid v \in V_{F^s}, p \in V_P\}$  are respectively the sets of loading/unloading arcs representing a departure/arrival at port  $p$  visited in  $F^s$ ,  $c_a = c_L^p$ , and  $c_a = c_U^p$  is the (un)loading cost for a container at the associated port  $p \in V_P$ ,  $t_a = 0$ , and  $C_a$  is unlimited.
- $A_T = \{(v, u) \mid v \in V_{F^s}, u \in V_{F^{s'}}\}$  is the set of transshipment arcs representing a transshipment between services  $F^s$  and  $F^{s'}$  defined for every pair  $(v, u)$  where  $p(v) = p(u)$ ,  $c_a = c_T^p$  is the transshipment cost for a container at the associated port  $p \in V_P$ ,  $t_a$  is the transshipment time, and  $C_a$  is unlimited.
- $A_S = \{(v, u) \mid s \in S, v, u \in V_{F^s}, v = p_h^s, u = p_{((h+1) \bmod m)}^s\}$  is the set of sailing arcs representing a sailing between two consecutive port calls  $v$  and  $u$  in  $F^s$ ,  $c_a = 0$  as sailing costs are directly incurred by the vessels,  $t_a = d_{uv}/v_s + b_v$  meaning the time in hours to traverse the edge plus the berthing time at the arriving port for each sailing, and  $C_a = U_{e(s)}$ .
- $A_K = \{(v, u) \mid v, u \in V_P, \exists k \in K : O_k = v \wedge D_k = u\}$  is the set of forfeiting arcs representing a rejection of transporting the cargo  $k$  between  $v$  and  $u$  in  $P$ ,  $c_a = \tilde{z}_k + z_k$  is the penalty associated with rejecting the cargo  $k$ ,  $t_a = t_k$  is the maximum transit time for  $k$ , and  $C_a = q_k$ .

We use the path flow formulation of the time constrained MCF problem as described in Karsten et al. (2015). Let  $\Omega_k$  be the set of all feasible paths for commodity  $k$  including forfeiting the cargo. Let  $\Omega(a)$  be the set of all paths using arc  $a \in A$ . The cost of a path  $\rho$  is denoted as  $c_\rho$  and it includes the revenue obtained by transporting one unit of commodity  $k$  sent along path  $\rho \in \Omega_k$ . The real nonnegative variable  $x_\rho$  denotes the amount of commodity  $k$  sent along the path. Let the weekly cost of a service be  $c_s = n_{e(s)}f_{e(s)} + \sum_{(i,j) \in A_S} (c_B(h^{e(s)}b_p + g_{v(s)}^{e(s)}d_{ij}^{e(s)}) + c_j^{e(s)} + l_{ij}^{e(s)})$  accounting for fixed cost of deploying the vessel and the variable cost in terms of the fuel and port call cost of one round trip. Define binary service variables  $y_s$  indicating the inclusion of service  $s \in S$  in the solution.

Then the TCLSNBP can be formulated as the following mixed integer program:

$$\min \quad \sum_{s \in S} c_s y_s + \sum_{k \in K} \sum_{\rho \in \Omega_k} c_\rho x_\rho \quad (4.1)$$

$$\text{s.t.} \quad \sum_{\rho \in \Omega_k} x_\rho = q_k \quad k \in K \quad (4.2)$$

$$\sum_{\rho \in \Omega(a)} x_\rho - U_{e(s)} y_s \leq 0 \quad s \in S, a \in A_S \quad (4.3)$$

$$\sum_{s \in S: e(s)=e} n_{e(s)} y_s \leq N_e \quad e \in E \quad (4.4)$$

$$x_\rho \in \mathbb{R}^+ \quad \rho \in \Omega_k, k \in K \quad (4.5)$$

$$y_s \in \{0, 1\} \quad s \in S \quad (4.6)$$

The objective (4.1) minimizes cumulative service and cargo transportation cost. As the cargo transportation cost includes the revenue of transporting the cargo this is equivalent to maximizing profit. *The cargo flow constraints* (4.2) along with non-negativity constraints (4.5) ensure that all cargo is either transported or forfeited. *The capacity constraints* (4.3) link the cargo paths with the service capacity installed in the transportation network. *The fleet availability constraints* (4.4) ensure that the selected services can be operated by the available fleet. Finally, constraints (4.6) define the service variables as binary.

The mathematical model extends the problem description of the LSNBP presented in Brouer et al. (2014a) to handle transit times. The model enforces a weekly frequency resulting in a weekly planning horizon. The path flow formulation of the MCF problem considers transit time restrictions in the definition

of a feasible path for a given commodity. Column generation is applied for solving the path flow formulation of the MCF problem, where reduced cost columns are generated by solving a shortest path problem. Introducing transit time restrictions changes the sub-problem to a resource constrained shortest path problem and thus the complexity of the sub-problem becomes NP-hard. The label setting algorithm from Karsten et al. (2015) is used to solve the cargo routing problem with transit time restrictions during the execution of our algorithm.

In the TCLSNDP the sailing speed is decisive for the cost of a given service as well as the feasible solution space of the multicommodity flow problem. The majority of all commodities are subject to transshipments and transit time may depend on the choice of speed on multiple services. As a consequence lowering the speed to reduce the cost of a service may make existing cargo routings infeasible due to an increase in transit times. Likewise, increasing speed may result in increased flow in the network as the set of feasible paths increase, but at the same time it will increase the cost of service through the additional fuel burn. The service variables of (4.1)-(4.6) are defined for an average speed on all sailings on a round trip and assume a fixed weekly frequency. Hence, the resulting speed and cost change from in- or de-creasing by one vessel may be quite significant. However, the proposed algorithm does not optimize speeds of the individual sailings. The feasible deployment of vessels to maintain weekly frequency will be limited by the minimum and maximum speed.

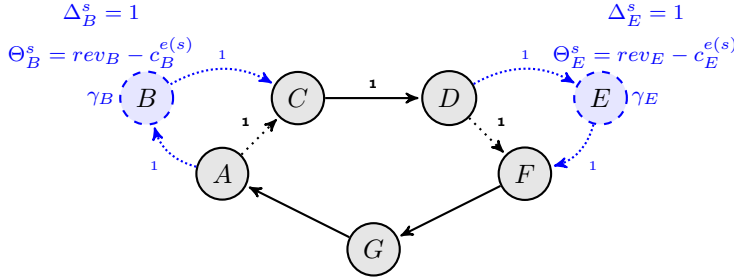
### 4.3 Algorithm

The algorithm presented in this paper is an extension of the matheuristic for the LSNDP presented in Brouer et al. (2014b). The algorithm proposed in Brouer et al. (2014b) uses a greedy knapsack based construction heuristic to create an initial set of services,  $S$ . Then the core of the matheuristic is executed iteratively to try to improve these using a MIP for each service. The algorithm terminates either when no profitable moves can be found or when a computational time limit is reached. We use the same overall framework in the following and a detailed description and flow chart of the algorithm can be found in Brouer et al. (2014b). The central component in the latter matheuristic is an improvement heuristic, where an integer program is solved as a move operator in a large neighborhood search. The integer program is iteratively solved for a single service using estimation functions for changes in the flow due to insertions and removals of port calls in the service investigated. The solution of the integer program provides a set of moves in the composition of port calls and

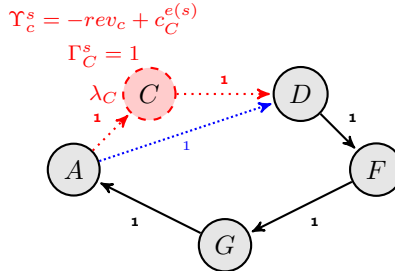
fleet deployment. Flow changes and the resulting change in the revenue are estimated by solving a series of resource constrained shortest path problems on the residual graph of the current network. This is done for relevant commodities to the insertion/removal of a port call along with an estimation of the change in the vessel related cost with the current fleet deployment.

Given a total estimated change in revenue of  $rev_i$  and port call cost of  $c_i^{e(s)}$  Figure 4.1a illustrates estimation functions for the change in revenue ( $\Theta_i^s$ ) and duration increase ( $\Delta_i^s$ ) for inserting port  $i$  into service  $s$  controlled by the binary variable  $\gamma_i$ . The duration controls the number of vessels needed to maintain a weekly frequency of service. Figure 4.1b illustrates the estimation functions for the change in revenue ( $\Upsilon_i^s$ ) and decrease in duration ( $\Gamma_i^s$ ) for removing port  $i$  from service  $s$  controlled by the binary variable  $\lambda_i$ . Insertions/removals will affect the duration of the service in question and hence the needed fleet deployment modeled by the integer variable  $\omega_s$  representing the change in the number of vessels deployed.

(a) Blue nodes are evaluated for insertion corresponding to variables  $\gamma_i$  for the set of ports in the neighborhood  $N^s$  of service  $s$ .



(b) Red nodes are evaluated for removal corresponding to variables  $\lambda_i$  for the set of current port calls  $F^s$  on service  $s$ .

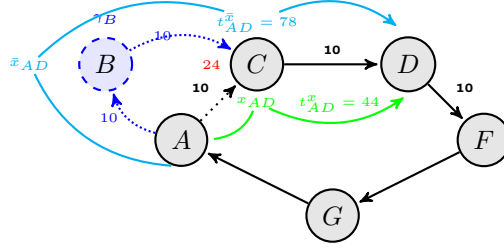


**Figure 4.1:** The estimation functions for insertion and removal of port calls.



### 4.3.1 Revenue Loss due to Transit Time Changes

For considering the transit time in the IP, it is necessary to estimate how insertions and removals of port calls will affect the duration of the existing flow on the service. This means that existing flow must be estimated to have sufficient slack in transit time for the insertions performed or alternatively, existing flow will result in a loss of revenue if it cannot be rerouted within the available transit time on a different path. Figure 4.2 illustrates a case of a path with the associated variable in the current basis of the MCF model. When inserting port  $B$  on its path it becomes infeasible due to transit time restrictions.



**Figure 4.2:** Insertions/removals affect transit time of the flow. Commodity  $k_{AD}$  has a maximum transit time of 48 hours and the insertion of port call  $\gamma_B$  will make the path infeasible.

In order to account for the transit time restrictions of the current flow, constraints (4.14) are added to the IP and a penalty,  $\zeta_x$  corresponding to losing the cargo, is added to the objective if the transit time slack for an existing path becomes negative. This is handled through the variable  $\alpha_x$ , where  $x$  refers to a path variable with positive flow in the current solution and  $s_x$  refers to the current slack time according to the transit time restrictions of the associated path. For ease of reading, Table 4.1 gives an overview of additional sets, constants, and variables used in the IP.

Sets	
$N^s$	Set of neighbors (potential port call insertions) of $s$ .
$X^s$	Set of path variables on service $s$ in current flow solution with pos. flow.
$N^x \subseteq N^s$	Subset of neighbors inserted in current path of variable $x \in X^s$ .
$F^x \subseteq F^s$	Subset of port calls on current path of variable $x \in X^s$ .
$L_i$	Lock set for port call insertion $i \in N^s$ or port call removal $i \in F^s$ .
Constants	
$Y_s$	Distance of the route associated with $s$ .
$K_s$	Estimated average speed of the service $s$ .
$M_{e(s)}$	Number of undeployed vessels of class $e$ in the current solution.
$I_s$	Maximum number of insertions allowed in $s$ .
$R_s$	Maximum number of removals allowed in $s$ .
$\Delta_i^s$	Estimated distance increase if port call $i \in N^s$ is inserted in $s$ .
$\Gamma_i^s$	Estimated distance decrease if port call $i \in F^s$ is removed from $s$ .
$\Theta_i$	Estimated profit increase of inserting port call $i \in N^s$ in $s$ .
$\Upsilon_i$	Estimated profit increase of removing port call $i \in F^s$ from $s$ .
$\zeta_x$	Estimated penalty for cargo lost due to transit time.
$s_x$	Slack time of path variable $x$ .
Variables	
$\lambda_i$	1 if port call $i \in F^s$ is removed from $s$ , 0 otherwise.
$\gamma_i$	1 if port call $i \in N^s$ is inserted in $s$ , 0 otherwise.
$\omega_s \in \mathbb{Z}$	Number of vessels added (removed if negative) to $s$ .
$\alpha_x$	1 if transit time of path variable $x \in X^s$ is violated, 0 otherwise.

**Table 4.1:** Overview of sets, constants, and variables used in the IP.

Given this notation, the IP is:

$$\max \sum_{i \in N^s} \Theta_i \gamma_i + \sum_{i \in F^s} \Upsilon_i \lambda_i - f_{e(s)} \omega_s - \zeta_x \alpha_x \quad (4.7)$$

Subject to:

$$\frac{Y_s}{K_s} + \sum_{i \in F^s} b_{p(i)} + \sum_{i \in N^s} \left( \frac{\Delta_i^s}{K_s} + b_{p(i)} \right) \gamma_i - \sum_{i \in F^s} \left( \frac{\Gamma_i^s}{K_s} + b_{p(i)} \right) \lambda_i \leq 168 \cdot (n_{e(s)} + \omega_s) \quad (4.8)$$

$$\omega_s \leq M_{e(s)} \quad (4.9)$$

$$\sum_{i \in N^s} \gamma_i \leq I_s \quad (4.10)$$

$$\sum_{i \in F^s} \lambda_i \leq R_s \quad (4.11)$$

$$\sum_{j \in L_i} \lambda_j \leq |L_i| (1 - \gamma_i) \quad i \in N^s \quad (4.12)$$

$$\sum_{j \in L_i} \lambda_j \leq |L_i| (1 - \lambda_i) \quad i \in F^s \quad (4.13)$$

$$\begin{aligned}
& \sum_{i \in N^x} \left( \frac{\Delta_i^s}{K_s} + b_{p(i)} \right) \gamma_i - \sum_{i \in F^x} \left( \frac{\Gamma_i^s}{K_s} + b_{p(i)} \right) \lambda_i - UB \alpha_x \leq s_x \quad x \in X^s \quad (4.14) \\
& \lambda_i \in \{0, 1\}, \quad i \in F^s \quad \gamma_i \in \{0, 1\}, \quad i \in N^s \quad \alpha_x \in \{0, 1\}, \quad x \in X^s \\
& \omega_s \in \mathbb{Z}, \quad s \in S
\end{aligned}$$

The objective function (4.7) maximizes the estimated profit increase obtained from removing and inserting port calls, accounting for the estimated change of revenue, transshipment cost, port call cost and fleet cost. As opposed to the IP proposed in Brouer et al. (2014b) the change in revenue may be related to not transporting cargo for which the path duration is estimated to exceed the transit time of the commodity. The number of vessels needed on the service (assuming a weekly frequency) after insertions/removals is estimated in constraint (4.8) accounting for the change in the service time given the current speed  $K_s$ . Constraint (4.9) ensures that the solution does not exceed the available fleet of vessels. Note that  $\omega_s$  does not need to be bounded from below by  $-n_{e(s)}$  because it is not allowed to remove all port calls. Constraints (4.10) and (4.11) limit the number of port call insertions and removals to minimize the error in the computed estimates. The set of port calls affected by an insertion or a removal is fixed by the lock set constraints (4.12) and (4.13), respectively. Finally, constraints (4.14) activate the estimated penalty for lost cargo due to an estimated violation of the transit time for the commodity on this particular path.

## 4.4 Computational Results

The matheuristic was tested on the benchmark suite *Liner-lib* described in Brouer et al. (2014a). The instances can be found at <http://www.liner-lib.org>. Table 4.2 gives an overview of the instances. We have revised the transit time restrictions for a small number of the origin-destination pairs in order to meet critical transit times as our model operates with average sailing speeds. The pairs where the transit times have been revised are those that cannot be satisfied by a direct sailing at 14 knots. The number of revised pairs is 6, 15, 106, and 32 for WAF, Pacific, WorldSmall, and AsiaEurope respectively. They have been revised according to the most recent published liner shipping transit times.

The matheuristic has been coded in C++ and run on a linux system with an *Intel(R) Xeon(R) X5550* CPU at *2.67GHz* and *24 GB* RAM. The algorithm is set to terminate after the time limits imposed in Brouer et al. (2014a).

	Instance and description	$ P $	$ K $	$ E $	min $v$	max $v$	t.l.
<b>Single-hub</b>	<b>Baltic</b> Baltic sea, Bremerhaven as hub	12	22	2	5	7	300
	<b>WAF</b> West Africa, Algeciras as hub	19	38	2	33	51	900
<b>Multi-hub</b>	<b>Mediterranean</b> Mediter-ranean, Algeciras, Tangier, and Gioia Tauro as hubs	39	369	3	15	25	1,200
<b>Trade-lane</b>	<b>Pacific</b> (Asia-US West)	45	722	4	81	119	3,600
	<b>AsiaEurope</b> Europe, Middle East and Far east regions	111	4,000	6	140	212	14,400
<b>World</b>	<b>Small</b> 47 Main ports worldwide identified by Maersk Line	47	1,764	6	209	317	10,800

**Table 4.2:** The instances of the benchmark suite with indication of the number of ports ( $|P|$ ), the number of origin-destination pairs ( $|K|$ ), the number of vessel classes ( $|E|$ ), the minimum and maximum number of vessels (min  $v$  and max  $v$ ), and the solution time limit in seconds (t.l.).

We fix the berthing time,  $b_p$ , to 24 hours for all ports as in Brouer et al. (2014a) and the transshipment time,  $t_a$ , is fixed to 48 hours for every connection. If a schedule was considered the inter-service transshipment time could be calculated based on arrival and departure times.

#### 4.4.1 Computational Results for *Liner-lib*

Table 4.3 shows that the algorithm can find profitable solutions (negative objective values) for Baltic, WAF, WorldSmall and AsiaEurope. Pacific is unprofitable although both fleet deployment and transport percentage is high. In most instances except the Mediterranean around 85 % to 95 % of the available cargo is transported on average. At the same time as little as 80 % of the fleet in terms of volume is utilized suggesting that further improvements may be achievable as the larger instances all terminate due to the imposed computational time limits. For the smaller instances the stopping criterion of the embedded simulated annealing procedure is fulfilled before the time limit is reached.

Table 4.4 shows that most services operate relatively close to their design speed for the smaller classes, apart from the WorldSmall instances where average ser-

Instance	Objective	Deployment		Flow	CPU time
	$\mathbf{Z(7)}$	$\mathbf{D(v)}$ (%)	$\mathbf{D( E )}$ (%)	$\mathbf{T(v)}$ (%)	$\mathbf{(S)}$
<b>Best - Baltic</b>	<b>-14,050</b>	<b>100.0</b>	<b>100.0</b>	<b>87.4</b>	<b>101</b>
Average - Baltic	74,480	100.0	100.0	86.7	108
<b>Best - WAF</b>	<b><math>-5.59 \cdot 10^6</math></b>	<b>83.3</b>	<b>85.7</b>	<b>97.0</b>	<b>255</b>
Average - WAF	$-4.87 \cdot 10^6$	83.3	85.2	94.3	354
<b>Best - Med</b>	<b><math>2.42 \cdot 10^6</math></b>	<b>91.9</b>	<b>95.0</b>	<b>86.9</b>	<b>710</b>
Average - Med	$2.70 \cdot 10^6$	90.5	94.0	78.9	737
<b>Best - Pacific</b>	<b><math>3.81 \cdot 10^6</math></b>	<b>95.5</b>	<b>96.0</b>	<b>94.7</b>	<b>time</b>
Average - Pacific	$4.62 \cdot 10^6$	92.2	90.0	92.1	time
<b>Best - WorldSmall</b>	<b><math>-3.18 \cdot 10^7</math></b>	<b>81.2</b>	<b>89.4</b>	<b>90.7</b>	<b>time</b>
Average - WorldSmall	$-2.80 \cdot 10^7$	81.9	88.4	90.2	time
<b>Best - AsiaEurope</b>	<b><math>-1.76 \cdot 10^7</math></b>	<b>87.1</b>	<b>95.5</b>	<b>90.7</b>	<b>time</b>
Average - AsiaEurope	$-1.45 \cdot 10^7$	84.8	92.3	88.4	time

**Table 4.3:** Best and Average of 10 runs. Weekly profit (objective value)  $\mathbf{Z(7)}$ ; percentage of fleet deployed: as a percentage of the total volume  $\mathbf{D(v)}$ , and as a percentage of the number of ships  $\mathbf{D(|E|)}$ .  $\mathbf{T(v)}$  is the percentage of total cargo volume transported and  $\mathbf{(S)}$  is the execution time in CPU seconds; time means the solution time limit given in Table 4.2 has been reached.

Instance	F450	F800	P1200	P2400	Post P	Super P
Baltic	10.8	13.7				
WAF	11.5	13.2				
Med	11.9	13.7	13.9			
Pacific	11.6	14.3	16.4	17.9		
WorldSmall	12.9	15.2	17.3	19.4	19.4	17.1
AsiaEurope	11.6	13.9	16.8	17.9	19.4	17.6
<b>Design Speed</b>	<b>12.0</b>	<b>14.0</b>	<b>18.0</b>	<b>16.0</b>	<b>16.5</b>	<b>17.0</b>
<b>Max speed</b>	<b>14.0</b>	<b>17.0</b>	<b>19.0</b>	<b>22.0</b>	<b>23.0</b>	<b>22.0</b>

**Table 4.4:** Average speed per vessel class over ten runs. Last two rows indicate the design speed and max speed of the corresponding vessel classes. F is Feeder, P is Panamax.

vice speed is higher than design speed. The larger Panamax vessel classes generally have high average speeds. For the WorldSmall and AsiaEurope, we can see in Table 4.3 that we have excess fleet and by comparing  $\mathbf{D}(\mathbf{v})$  and  $\mathbf{D}(|E|)$  it can be seen that it is mainly the large vessel classes that are undeployed. This is somewhat surprising as this contradicts the economy of scale of larger vessels. However, Table 4.4 also shows that the WorldSmall and AsiaEurope operate at very high speeds for the large vessel classes. An explanation could be the fact that we cannot swap vessel classes very well in the algorithm and we are perhaps not able to fill the larger vessels because we have very good utilization on the small services. This needs further investigation.

Instance		$ \mathbf{R} $	$\mathbf{tt}(\mathbf{k})$ (%)	$\mathbf{C}(\mathbf{k})$ (%)	$\mathbf{tt},\mathbf{C}(\mathbf{k})$ (%)	$\mathbf{L}(\mathbf{k})$ (%)	$\mathbf{FFE}$	$\mathbf{tt}(\mathbf{v})$ (%)	$\mathbf{C}(\mathbf{v})$ (%)	$\mathbf{tt},\mathbf{C}(\mathbf{v})$ (%)	$\mathbf{L}(\mathbf{v})$ (%)
Baltic	$\mu$	10	0.0	20.8	0.0	79.2	653	0.0	66.4	0.0	33.6
	$\sigma$	1	0.0	6.7	0.0	6.7	57	0.0	6.8	0.0	6.8
WAF	$\mu$	7	3.4	16.2	0.0	80.4	489	2.8	28.1	0.0	69.1
	$\sigma$	2	7.4	12.5	0.0	9.3	230	6.9	28.4	0.0	26.1
Med	$\mu$	113	32.9	0.7	5.1	61.2	1,590	41.9	0.7	7.4	50.0
	$\sigma$	25	11.1	0.9	3.1	12.2	521	13.0	1.2	5.1	14.6
Pacific	$\mu$	197	48.7	6.7	18.2	26.3	3,500	35.6	25.8	31.2	7.4
	$\sigma$	20	10.3	3.1	10.3	7.9	975	11.9	12.9	17.9	3.5
WorldSmall	$\mu$	258	37.1	31.4	17.6	13.9	12,591	41.3	26.7	22.9	9.1
	$\sigma$	37	12.2	8.3	7.1	14.9	2,060	13.3	6.9	9.4	9.6
AsiaEurope	$\mu$	823	37.7	9.8	23.4	29.2	8,918	44.4	14.9	26.5	14.2
	$\sigma$	144	9.7	4.6	5.8	7.1	1,828	13.8	7.6	8.9	4.8

**Table 4.5:** Statistics on the rejected demand reporting average ( $\mu$ ) and standard deviation ( $\sigma$ ) over ten runs.  $|\mathbf{R}|$  is the number of rejected OD pairs;  $\mathbf{tt}(\mathbf{k})$  is the percentage of OD pairs rejected due only to transit time;  $\mathbf{C}(\mathbf{k})$  is the percentage of OD pairs rejected due only to lack of capacity;  $\mathbf{tt},\mathbf{C}(\mathbf{k})$  is the percentage of OD pairs rejected due to both transit time and lack of capacity;  $\mathbf{L}(\mathbf{k})$  is the percentage of OD pairs not connected;  $\mathbf{FFE}$  is the volume of the rejected demand;  $\mathbf{tt}(\mathbf{v})$  is the percentage of the volume rejected due only to transit time;  $\mathbf{C}(\mathbf{v})$  is the percentage of the volume rejected due only to lack of capacity;  $\mathbf{tt},\mathbf{C}(\mathbf{v})$  is the percentage of volume rejected due to both transit time and lack of capacity;  $\mathbf{L}(\mathbf{v})$  is the percentage of volume rejected because O and D are not connected.

Table 4.5 gives statistics on the rejected demand in the solutions. The primary causes are that existing paths do not meet transit time restrictions, that there is no residual capacity or that the OD pair is not connected in the graph. For the smaller instances (Baltic, WAF, and Mediterranean) rejection of demand is primarily because the OD pairs are not connected, indicating that it is unprof-

Instance	tt (days)	Objective	Deployment		Cargo flow	CPU time
		$\mathbf{Z}(\mathbf{7})$	$\mathbf{D}(\mathbf{v})$ (%)	$\mathbf{D}( E )$ (%)	$\mathbf{T}(\mathbf{v})$ (%)	$(\mathbf{S})$
WorldSmall	+0 days	$-2.80 \cdot 10^7$	81.9	88.4	90.2	3
WorldSmall	+1 days	$-3.06 \cdot 10^7$	85.3	90.2	92.0	3
WorldSmall	+2 days	$-3.29 \cdot 10^7$	88.0	91.3	92.7	3
WorldSmall	+3 days	$-3.43 \cdot 10^7$	86.4	90.2	92.9	3
WorldSmall	+4 days	$-3.69 \cdot 10^7$	86.3	91.1	93.3	3
WorldSmall	+5 days	$-3.62 \cdot 10^7$	89.0	93.0	94.3	3
WorldSmall	unlimited (RCSP)	$-3.17 \cdot 10^7$	92.7	94.1	94.0	time
WorldSmall	unlimited (Dijkstra)	$-3.51 \cdot 10^7$	91.7	93.3	94.3	time

**Table 4.6:** Influence of extended transit time restrictions for all cargoes. Average of 10 runs. The additional number of days allowed for each cargo (**tt**). Weekly profit (objective value)  $\mathbf{Z}(\mathbf{7})$ ; percentage of fleet deployed: as a percentage of the total volume  $\mathbf{D}(\mathbf{v})$ , and as a percentage of the number of ships  $\mathbf{D}(|E|)$ .  $\mathbf{T}(\mathbf{v})$  is the percentage of total cargo volume transported and  $(\mathbf{S})$  is the CPU execution time in hours.

itable to call these ports. For the larger instances (Pacific, WorldSmall, and AsiaEurope) the demand is primarily rejected due to the transit times that cannot be met (with some variation), and in WorldSmall a significant amount of cargo is also rejected due to lack of capacity. In general comparing the percentage not connected in number of demands ( $k$ ) compared to the volume ( $v$ ) not connected indicates that it is the demands with low volume that are not connected. Often these demands are from small feeder ports not visited by the solution because the total volume is very low and it is deemed unprofitable by the algorithm.

## 4.5 Sensitivity

Table 4.6 shows the influence of increasing the transit time restrictions given in the *Liner-lib* with up to five days for all cargoes. The results show that the transit time restrictions imposed for each individual cargo are critical. In practice it will not be acceptable to extend all cargo transit times with many days, but as seen even an extension of two days increase the profit of the generated networks by roughly 10 %. The results also illustrate how automatic decision

support tools can help quickly assess the influence of the offered level of service. Likewise, it can serve as a real-time decision support tool to find improvements when evaluating changes in operating conditions or testing different scenarios based on e.g. different service parameters. In addition to the results with an additional five day allowance on the transit time, we have tested the algorithm with unlimited transit time in two settings. When there is no restriction on the transit time, it is possible to solve the shortest path problem using both the resource constrained shortest path, RCSP, algorithm and the Dijkstra algorithm. The performance of the overall algorithm when the Dijkstra algorithm is used is superior, as the sub-problem is solved more efficiently leading to more evaluations and consequently a better network. As seen there is generally significant savings by extending the allowed transit time. Interestingly, the results also show that solving the cargo routing problem with the limits on transit time relaxed by four or five days leads to more profitable networks than when considering unlimited transit time within the allowed computational running time as the shortest path problem is solved faster when transit time is tightly limited. The solution space of the cargo routing problem is limited sufficiently to make the algorithm converge faster than when solving the unrestricted problem, which is in accordance with Karsten et al. (2015).

The left part of Table 4.7 shows the performance of the algorithm for different running times. The results underline that when the algorithm is used to design a new network from scratch, the solution does improve by 10-20 % for the larger instances when running for extended periods, but close inspection of the results also show that the improvements are tailing off. Since the algorithm is intended as an improvement heuristic in practice, the results will be different here. When used as a decision support tool to make incremental changes of a network already in place, the initial solution (the existing network) will be of higher quality than the initial networks considered here, which are constructed by a simple heuristic.

## 4.6 Time and Transshipment Constrained Network Design

The transshipment operations are both expensive and time consuming as they require to unload, store, and re-load the container at the intermediate port. Additionally, having fewer transshipments generally reduces handling time, possibly transit time, the risk of damage, and also the risk of missing connections, which is beneficial from a customer perspective. Consequently, transporta-



Instance	Transit time restrictions					Transit time restrictions and limited transshipments				
	Cargo CPU					Cargo CPU				
	Obj.	Deployment	flow		time	Obj.	Deployment	flow		time
	<b>Z(7)</b>	<b>D(v)</b>	<b>D(E)</b>	<b>T(v)</b>	<b>(S)</b>	<b>Z(7)</b>	<b>D(v)</b>	<b>D(E)</b>	<b>T(v)</b>	<b>(S)</b>
	(M\$)	(%)	(%)	(%)	(h)	(M\$)	(%)	(%)	(%)	(h)
Pacific	4.62	92.2	90.0	92.1	1	5.10	92.2	88.7	91.0	1
Pacific	3.68	95.0	94.0	93.8	4	3.73	95.3	82.0	93.4	4
WorldSmall	-28.0	81.9	88.4	90.2	3	-25.8	85.2	91.3	90.4	3
WorldSmall	-31.0	84.4	90.3	91.5	12	-31.1	85.5	91.2	91.6	12
AsiaEurope	-14.5	84.8	92.3	88.4	4	-9.80	85.6	91.9	86.7	4
AsiaEurope	-17.7	86.9	93.4	89.9	16	-14.4	86.2	93.0	88.7	16

**Table 4.7:** Influence of extended CPU running time and additional level of service requirements. The level of service is considered in terms of restrictions on transit time (left) and restrictions on transit time and the number of transshipments limited to two (right). Average of 10 runs. Weekly profit in million dollars (Obj.) **Z(7)**; percentage of fleet deployed: as a percentage of the total volume **D(v)**, and as a percentage of the number of ships **D(E)**. **T(v)** is the percentage of total cargo volume transported and **(S)** is the CPU execution time in hours.

tions with fewer transshipments are preferred and some customers may actually specify a maximum number of transshipments for their containers, especially for hazardous and high-value cargo. From a carrier perspective, having fewer transshipments also reduces the non-value adding steps and handling capacity requirements, but transshipment operations are important since they permit using vessel capacities more effectively.

The path flow formulation of the multi-commodity flow problem used to solve the cargo routing problem can easily be extended to also explicitly limit the number of transshipments by introducing a resource that keeps track of these. Here, we extend the algorithm for solving the cargo routing problem to include restrictions on the number of transshipments by including an additional resource. The additional resource is incremented by one at transshipment edges and included in the dominance criterion in addition to transit time. Furthermore, paths that violate either the transshipment or transit time criterion are deleted.

Table 4.7 reports results for the network design algorithm, and as seen the solutions are still profitable when including a limit on the number of transshipments in addition to the transit time requirement. We impose that all cargoes cannot transship more than twice as this is the case in practice for

more than 95 % of the cargo. However, for some cargoes this will constrain the solution space too much, but we do not have more detailed information on the number of allowed transshipments available. The results for the shorter running times show that some cargo that was previously transported is now rejected and the objective values are approximately 10 % worse for Pacific and WorldSmall, whereas they are approximately 30 % worse for AsiaEurope. Further analysis reveals that solving the cargo routing problem with tight limits on transit time, and the number of transshipments limited to two, in terms of computational time is comparable to or slightly faster than when only considering transit time. So even though the shortest path algorithm is extended with an additional resource, the solution space of the cargo routing problem is limited sufficiently to make the algorithm converge faster. Hence, the level of detail considerations can help restrict the solution space sufficiently to offset the increased complexity of the label setting algorithm and turns out to improve dominance significantly. However, on the input side the number of transshipments allowed for each cargo will influence the performance. So even though the solution time of the sub-problem does not increase, it is harder for the network design algorithm to find good solutions. When we increase the allowed running time of the algorithm the picture changes as seen in Table 4.7. The profitability of the created networks with a better level of service is the same as the networks where only transit time is restricted. For the WorldSmall instance the profit of the network created only with transit time restrictions is 31 M\$, while the profit is 31.1 M\$ when considering both transit time restrictions and limited transshipments. This is a very encouraging result. We only show results for larger instances where some of the cargo is transshipping more than once (for the smaller instances all cargo is transported using direct or one-transshipment connections). Furthermore, Table 4.7 illustrates that transit time and number of transshipments are correlated, and generally limiting one of them will implicitly bound the other, e.g. a container that has a transit time requirement of one week can only transship very few times before the allowed transit time is violated. Likewise, it is only possible to reach a limited part of the network if a customer only allows one transshipment. In this case the transit time for a container will be bounded by the longest of the possible direct and one-transshipment connections. The ability to control the number of transshipments gives a way of offering products to the customers that are more tailored to their needs. Being able to offer fewer transshipments may attract new customers even though these products are offered at a higher price and it will be possible during the network design process to consider differentiated products between any pair of ports in terms of offered transit time and number of transshipments. Likewise, it is easy to get an overview of possible routings with more transshipments that can be offered at a lower price while helping to ensure a better utilization of the network.

## 4.7 Conclusions

We have presented a model for the TCLSNBP introducing transit time restrictions on each individual commodity while maintaining the ability to transship between services. We have extended the matheuristic of Brouer et al. (2014b) to handle the new constraints. The core component of the matheuristic is an integer program considering a set of removals and insertions to a service. We extend the integer program to consider how removals and insertions influence the transit time of the existing cargo flow on the service. Each iteration of the matheuristic provides a set of moves for the current set of services and fleet deployment, which lead to a potential improvement in the overall revenue. The evaluation of the cargo flow for a set of moves requires solving a time constrained multi-commodity flow problem, which we solve using column generation.

The introduction of transit time constraints changes the estimation functions for the improvement heuristic and the pricing problem of the column generation algorithm from an ordinary shortest path problem to a resource constrained shortest path problem. We apply the specialized label setting algorithm of Karsten et al. (2015) to achieve satisfactory performance. Additionally, we show that it is tractable to extend the algorithm to consider a limit on the number of allowed transshipments as we can still create profitable networks of comparable quality.

Extensive computational tests show that it is possible to generate highly profitable networks for the majority of the instances in *Liner-lib* when considering level of service requirements and, especially for the larger instances, the approach generates networks of good quality. Furthermore, the tests show that from a computational perspective it can even be advantageous to include level of service requirements. However, some demand is not served and the fleet is not utilized completely, especially for the larger vessel classes, suggesting that further algorithmic improvements may lead to even better solutions. In particular, we expect that speed optimization on individual legs as well as more flexibility in terms of possible vessel class swaps could improve the algorithmic performance.

## Acknowledgments

This project was supported by the Danish Maritime Fund under the “Competitive Liner Shipping Network Design” project.

## Bibliography

- Agarwal, R. and Ergun, O. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196.
- Álvarez, J. F. (2012). Mathematical expressions for level of service optimization in liner shipping. *Journal of the Operational Research Society*, 63(6):709–714.
- Archetti, C. and Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2:223–246.
- Brouer, B., Álvarez, J., Plum, C., Pisinger, D., and Sigurd, M. (2014a). A base integer programming model and benchmark suite for liner shipping network design. *Transportation Science*, 48(2):281–312.
- Brouer, B., Desaulniers, G., and Pisinger, D. (2014b). A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Gelareh, S., Nickel, S., and Pisinger, D. (2010). Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):991–1004.
- Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138.
- Liu, Z., Meng, Q., Wang, S., and Sun, Z. (2014). Global intermodal liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 61:28–39.
- Meng, Q., Wang, S., Andersson, H., and Thun, K. (2014). Containership routing and scheduling in liner shipping: Overview and future research directions. *Transportation Science*, 48(2):265–280.
- Mulder, J. and Dekker, R. (2014). Methods for strategic liner shipping network design. *European Journal of Operational Research*, 235(2):367–377.
- Notteboom, T. E. and Vernimmen, B. (2009). The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337.

- Plum, C., Pisinger, D., and Sigurd, M. M. (2014). A service flow model for the liner shipping network design problem. *European Journal of Operational Research*, 235(2):378–386.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374.
- Stopford, M. (2009). *Maritime Economics*. Routledge, third edition.
- Wang, S. and Meng, Q. (2014). Liner shipping network design with deadlines. *Computers and Operations Research*, 41(1):140–149.
- Wang, S., Meng, Q., and Sun, Z. (2013). Container routing in liner shipping. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):1–7.

# Simultaneous Optimization of Sailing Speed and Container Routing with Transit Time Restrictions

with S. Røpke and D. Pisinger<sup>1</sup>

---

## Abstract

We introduce a decision support tool for liner shipping companies to optimally determine the sailing speed and needed fleet for a global network. As a novelty we incorporate cargo routing decisions with tight transit time restrictions on each container such that we get a realistic picture of the utilization of the network. Furthermore, we show that it is possible to extend the model to include optimal time scheduling decisions such that the time associated with transshipments is also reflected accurately. To solve the speed optimization problem we propose an exact algorithm based on Benders decomposition and column generation that exploits the separability of the problem. Computational results show that the method is applicable to liner shipping networks of realistic size and that it is important to incorporate cargo routing decisions when optimizing speed.

---

<sup>1</sup>Karsten, C. V., Ropke, S., and Pisinger, D. (2015). Simultaneous Optimization of Container Ship Sailing Speed and Container Routing with Transit Time Restrictions. Submitted to: *Transportation Science*.

## 5.1 Introduction

Liner shipping companies operate a set of sailing routes to provide transport for containers so as to maximize their revenue. Once the strategic decisions of which markets to serve have been made by a carrier and the sailing routes have been determined, most companies will adjust the network continuously. This is done due to changes in the global economic environment such as fluctuations in fuel prices, freight rates, and container demand. One way of optimizing the profitability of the network is to minimize the cost related to the operation of the routes, the deployment of vessels, and the handling of cargo. However, Karsten et al. (2015) recently showed that this approach will likely result in prolonged transit times. From a customer perspective not only low cost but the *level of service* offered is of concern. The level of service represents both the transportation cost and the transit time provided for a given cargo. Therefore, among the most influential decisions is the sailing speed between the serviced ports and the deployment of the available fleet. Higher sailing speeds will offer better transit times to the customers but will at the same time be more expensive to operate as there is an inherent trade-off in operating a low cost network versus a competitive network which is optimized in terms of both cost and offered cargo transit times. In the longer perspective, changes in sailing speed will also affect strategic decisions regarding the required fleet size as sailing routes, usually called *rotations*, are cyclic and require a weekly frequency, i.e. for a route the number of vessels deployed will correspond to the number of weeks it takes one vessel to complete a round trip, which will vary greatly depending on the sailing speed. The sailing speed has a significant impact on the operating costs as bunker may constitute more than 75% of the total operating cost of a vessel (Ronen, 2011). Furthermore, the consumption is approximately cubic in speed. Therefore, it is important to have a model that accurately assesses the impact of changes in sailing speed both from an operational, tactical and strategic perspective.

As a novelty we integrate the problems of sailing speed optimization, fleet deployment, and time constrained cargo routing as the decisions are highly dependent. Our model and solution method is aimed at optimizing the sailing speed for all sailing legs or a subset of these in a global liner shipping transportation network so as to maximize profit. For each rotation the fleet deployment can be adjusted maintaining weekly frequency and the speed between any pair of serviced ports is selected from a discrete set of speeds based on the characteristics of the deployed vessel class. The model is solved using Benders decomposition (constraint generation) where the rows are generated by solving a time constrained multi-commodity flow problem using column generation. That way we select the optimal sailing speed for each sailing leg

under consideration of cargo transit time restrictions. This also means that by speeding up, new cargo that is not currently transported may become available to transport. Furthermore, we show in Appendix 5.8.1 that it is possible to extend the model to include optimal time scheduling decisions such that the port arrival and departure times, and the time associated with transshipments is also reflected accurately. The extended model makes it possible to determine an optimal time schedule and corresponding optimal sailing speeds while considering optimal routings of cargo subject to transit time restrictions.

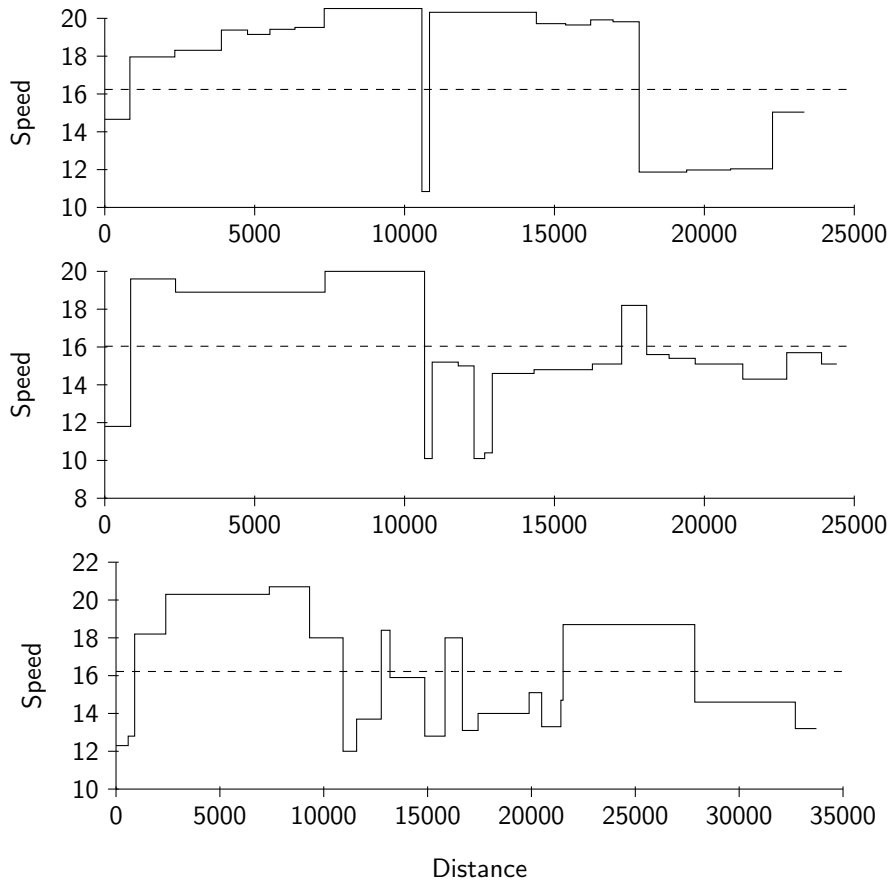
Christiansen et al. (2004, 2013) provide comprehensive reviews of the more recent literature on ship routing and scheduling. Brouer et al. (2014a) and Meng et al. (2014) give an introduction to the domain of liner shipping and an overview of recent literature specific to this area. The literature on optimization of liner shipping networks has been growing significantly during the last decade, and several planning problems at both the strategic, tactical and operational level have been addressed. Notteboom and Vernimmen (2009) and Ronen (2011) give background on speed optimization in liner shipping and show the importance of optimizing speed in liner shipping networks by studying a single rotation. Wang and Meng (2012c) formulate the speed optimization problem in a liner shipping network as a non-linear MIP. Cargo routing is considered for a pre-defined set of container routes where all demand must be met and cost is minimized in the model. Cheaitou and Cariou (2012) propose a model that explore and incorporate the available demands' dependence on transit time. Gelareh and Meng (2010) present a model for fleet deployment in a network where they also determine the sailing speed necessary to meet all demand while minimizing costs. Meng and Wang (2011) study the same problem for a single rotation. Similarly Zacharioudakis et al. (2011) optimize speed in a fleet deployment model, which they solve by assigning ships using a genetic algorithm. Xia et al. (2015) present a heuristic for optimizing fleet deployment and speed in an aggregated network but do not consider transshipments. They report computational results based on an aggregated network of up to 18 nodes. Psaraftis and Kontovas (2013) survey models and taxonomy on speed optimization in maritime transportation and Psaraftis and Kontovas (2015) discuss the practice of slow steaming. A related tactical problem is studied by Wang and Meng (2012a) who consider fleet deployment and transit time in a space time network and Wang and Meng (2012b) who study a tactical schedule model, where cost is minimized while maintaining a required transit time under uncertainty. Karsten et al. (2015) develop an efficient algorithm for the time-constrained cargo routing problem. This problem arises as a sub-problem in many of the tactical and strategic planning problems encountered by liner shipping companies when level of service is considered.



### 5.1.1 Industry Practice

The current practice used by major liner shipping companies is to vary speed across each of the operated rotations. Figure 5.1 shows the speed profile for three rotations recently operated by one of the leading global carriers, Maersk Line. It is clearly seen that speed is varied along the rotation and that it is rarely operated at or near the average speed for the entire rotation. The main driving factors in determining the sailing speed is the fuel price and whether the vessel is on its head or back haul, i.e. sailing in the cargo intensive direction or not. Most empirical findings as well as hydrodynamics suggest that the fuel consumption per time unit for container vessels is proportional to the third power of the sailing speed. In other words the fuel consumption per unit distance is proportional to the second power of the sailing speed. However, it is vessel dependent and the relationship can best be derived empirically. There is some evidence that for certain weather and hull conditions the bunker consumption can be greater than cubic in the speed, (Kontovas and Psaraftis, 2011) and, for large container vessels sailing at high speed, the power requirement may even be proportional to the fourth power of sailing speed (Man, 2013). For a fleet of vessels Wang and Meng (2012c) found the exponent to be between 2.7 and 3.3 empirically. In accordance with this, and following the benchmarks in Brouer et al. (2014a), we assume a third power relationship in the rest of this paper. For this relationship reducing speed by 20 % can give up to a 50 % reduction of fuel consumption and corresponding emissions for a vessel, or up to a 35 % reduction in fuel consumption for a rotation since it requires operating additional vessels in order to meet demand. However, some time critical transportation requests may not be available if operating at reduced speeds. As a consequence, lowering transportation cost while offering competitive cargo transit times (and a low number of transshipments) presents an inherent trade-off as fuel cost is the most important factor contributing to the operational cost of a network. This means that it may be worth selecting a more “expensive” rotation configuration which offers better connections. However, this may also save one vessel on the rotation. To address this we introduce a model to optimize the sailing speed and fleet deployment in a liner shipping network while considering a tight transit time restriction on each individual container. To solve the model we propose a decomposition based algorithm based on simultaneous column and row generation.

The rest of the paper is organized as follows. Section 5.2 introduces the needed transportation network. Section 5.3 describes the optimization problem and Section 5.4 shows the decomposition, derives stronger Benders cuts, and discusses how additional Benders cuts can be generated. Section 5.5 describes the solution algorithm. Computational results are presented in Section 5.6 for the

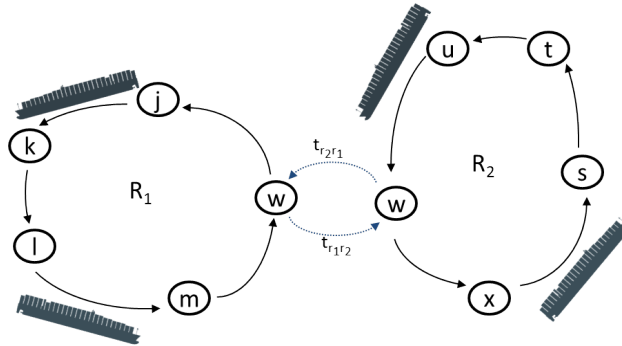


**Figure 5.1:** Speed profiles for three different rotations (AsiaEurope1, AsiaEurope10, AsiaEurope6TP6) operated by Maersk Line. The distance is in nautical miles and speed in knots. The steps corresponds to the average speed between waypoints (ports, canals etc.) on the rotation. Hence, some parts of the rotations might be operated at a lower or higher speed than showed. The dashed line is the average speed for the entire rotation.

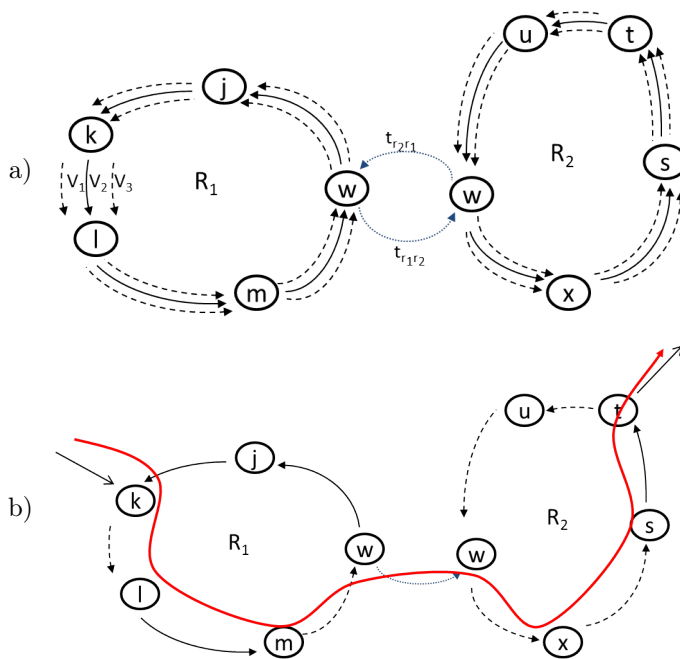
speed optimization problem before finally discussing possible extensions and concluding in Section 5.7. Appendix 5.8.1 shows how the model and solution method can be extended to include a time schedule. Appendix 5.8.2 discusses additional model improvements.

## 5.2 Transportation Network

Figure 5.2 illustrates a basic container shipping network. In this example the network is composed of two rotations  $R_1$  and  $R_2$  visiting various ports (nodes) and the solid black arcs correspond to sailing arcs. Containers can be transported between any pairs of ports and if the origin and destination port is not serviced by the same vessel, it can be transshipped between rotations at intermediate ports where rotations meet. In Figure 5.2 containers can be transshipped between rotation  $R_1$  and  $R_2$  in node  $w$  by using a transshipment arc which has an associated cost and time. Depending on the time schedule of the two rotations, the delay associated with the transshipment can be determined. In the following an exact time schedule is not known so an estimated transshipment time is used. Optimization with an actual time schedule is further addressed in Appendix 5.8.1. The capacity of each arc is determined by the size of the vessel deployed and the time it takes to traverse an arc by the sailing speed. As there are two vessels assigned to each rotation the total round trip time for each rotation must be two weeks to satisfy the weekly frequency requirement used by most liner shipping companies (Brouer et al., 2014a), but the speed on each sailing leg can vary greatly as discussed in the previous section. Figure 5.3a) shows an example of the transportation network we use in the model before speeds have been selected. We duplicate each sailing leg (dashed black lines in the figure) and assign different possible speeds to the duplicates such that cost and transit time is known a priori. E.g. between port  $k$  and  $l$  it is possible for the model to choose between three different speeds,  $V_1$ ,  $V_2$ , and  $V_3$ . Figure 5.3b) shows an example flow from  $k$  to  $t$  where the speeds for each leg have been selected. The sailing speed between e.g.  $k$  and  $l$  is selected at  $V_1$ . This way it is possible to calculate the transit time from  $k$  to  $s$  as the length of each sailing leg divided by the selected speed for each of the legs plus the average transshipment time. If the total transit time from  $k$  to  $s$  exceeds the requirement for a commodity going from  $k$  to  $s$ , there is no feasible path and it may be worth adjusting the speeds. If the speed is increased significantly on all legs in a rotation, it may be possible to reduce the number of vessels and still maintain weekly frequency. Likewise it may be necessary to use an extra vessel if speed is reduced significantly on all legs.



**Figure 5.2:** Simple representation of a liner shipping transportation network. The nodes correspond to ports and the arcs to sailing legs. In port  $w$  it is allowed to transship containers between rotation  $R_1$  and  $R_2$ .



**Figure 5.3:** Figure a) a transportation network used for speed optimization. Figure b) the flow through a network at a specific speed where load and unload arcs are included to correctly account for cost and time. The nodes correspond to ports and port  $w$  allows transshipments between rotations  $R_1$  and  $R_2$ .

### 5.3 Mathematical Modeling

To formulate the sailing speed optimization problem we formally define the graph described in the previous section  $G = (N, A)$  with nodes  $N$  and directed arcs  $A$ . The arcs in  $A$  are based on the original sailing arcs in an existing network,  $\bar{A}$ , at different speeds and hence  $A$  contains multiarcs. This is illustrated in Figure 5.3 a). Here multiple arcs are shown which are based on the original sailing arcs,  $\bar{A}$ , shown in Figure 5.2. Therefore, if, for example, every original sailing arc is considered at three different speeds, then  $|A| = 3|\bar{A}|$ , as illustrated by Figure 5.3 a). For important sailing legs the optimal sailing speed can be determined in greater detail (by considering more arcs) than at other sailing legs which are less flexible, e.g. because of fixed berthing times at both the departure and arrival port. To earn a revenue there is a set of commodities  $K$  that can be transported through the network between various origin-destination pairs. The amount of commodity  $k \in K$  that is available to be transported is  $d^k$ . W.l.o.g. we assume that each commodity has a single origin node and a single destination node. Furthermore, let  $q_a$  be the capacity of arc  $a \in A$  and  $t_a$  be the travel time for arc  $a \in A$  measured in days, including port time in the destination port. The decision variables  $x_a$  specify whether arc  $a \in A$  is used. The amount of commodity  $k \in K$  that is routed through path  $p$  is determined by  $y_p^k$ . The set of possible paths for commodity  $k$  is denoted  $P^k$  and the set of all paths is denoted  $P$ . Only paths that satisfy the given transit time restriction for commodity  $k$  are included. The integer decision variable  $L_r$  specifies the number of vessels used for rotation  $r$ . The set of rotations is denoted  $R$ . The set of rotations using vessel class  $v \in V$  is given by  $R(v)$ , and  $N^v$  specifies the number of available vessels of class  $v$ . The set of arcs that can be used by a rotation is denoted  $E(r) \subseteq A$ . The set  $P(a, k)$  contains the set of paths for commodity  $k \in K$  using arc  $a \in A$ . The multiarcs corresponding to a given sailing arc,  $\bar{a}$ , at different speeds are denoted  $A(\bar{a})$ . The cost of using arc  $a \in A$  is  $c_a$  and it includes the portion of the vessels fuel used at this arc sailing at the corresponding speed. Hence, the model easily allows different bunker consumption rates for different vessel classes and the non-linearity of the consumption as a function of speed is handled through the multiarcs. The cost of using a vessel at rotation  $r$  is  $C^r$  and the cost of sending commodity  $k$  through path  $p$  is  $r_p^k$ . A negative cost corresponds to a profitable path where a revenue can be obtained. The revenue includes loading, unloading, and transshipment costs and additionally there is a service penalty for not meeting demand. The costs are handled by introducing additional load, unload, and transshipment arcs as described in Karsten et al. (2015). Additionally, these arcs make sure we obtain the correct travel time for cargo through the network, including loading, unloading and transshipment time. We use the same objective (costs, revenues and penalties) as described in the reference model by Brouer et al. (2014a)

and a negative objective value indicates a profitable network. As we wish to maximize profit, this corresponds to minimizing the following objective in the integrated sailing speed optimization and cargo routing model, which is given by:

$$\min \sum_{a \in A} c_a x_a + \sum_{r \in R} C^r L^r + \sum_{k \in K} \sum_{p \in P^k} r_p^k y_p^k \quad (5.1)$$

subject to

$$\sum_{a \in A(\bar{a})} x_a = 1 \quad \bar{a} \in \bar{A} \quad (5.2)$$

$$\sum_{p \in P^k} y_p^k \leq d^k \quad k \in K \quad (5.3)$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq x_a q_a \quad a \in A \quad (5.4)$$

$$\sum_{a \in E(r)} t_a x_a \leq 7L_r \quad r \in R \quad (5.5)$$

$$\sum_{r \in R(v)} L_r \leq N^v \quad v \in V \quad (5.6)$$

$$y_p^k \in \mathbb{R}_+ \quad k \in K, p \in P^k \quad (5.7)$$

$$x_a \in \{0, 1\} \quad a \in A \quad (5.8)$$

$$L^r \in \mathbb{Z}_+ \quad r \in R \quad (5.9)$$

The objective (5.1) maximizes the total profit by minimizing the variable and fixed cost as well as the transportation cost (a negative transportation cost corresponds to a profitable path). Constraints (5.2) ensure that only one of the multiarcs at different speeds is selected such that a vessel is assigned exactly one speed at arc  $a$ . Constraints (5.3) assign cargo to paths to meet the demand or reject it if not profitable. Constraints (5.4) make sure that flow is only permitted on the selected arcs and the capacity of the arc is not violated. Finally, Constraints (5.5) and (5.6) ensure enough vessels are assigned to all rotations to meet the weekly frequency requirement without violating the fleet availability for each vessel class. Here the time  $t_a$  is measured in days and includes the port time in the origin port. If  $t_a$  is measured in hours the right-hand-side of (5.4) should be multiplied by the number of hours per week (168) rather than the days per week.

## 5.4 Decomposition

The model (5.1)-(5.9) is difficult to solve directly since it contains a large number of variables. The number of  $y_p^k$  variables can grow exponentially in the size of the graph. One solution approach would be to solve the LP relaxation of the model using column generation and obtain integer solutions using a branch-and-price algorithm (see e.g. Barnhart et al. (1998) for more information about branch-and-price).

Here we suggest a different approach. We notice that the  $y_p^k$  variables all are continuous. This means that we can apply Benders decomposition to model (5.1)-(5.9) and place the constraints related to the  $y_p^k$  variables in the sub-problem. The sub-problem in Benders decomposition has to be solved by column generation but the master problem can be solved either using a standard integer programming solver or using a branch and cut framework that allows the user to add cut callbacks. Both approaches are typically simpler to implement compared to a full branch-and-price algorithm. A Benders decomposition algorithm furthermore has the advantage that it continuously produces feasible solutions such that the method works as a heuristic when it is stopped before optimality is reached. A potential drawback is that algorithms based on Benders decomposition have a reputation of converging slowly.

In the following we are going to review the parts of Benders decomposition algorithm that are necessary for our application. The presentation is largely based on Costa (2005). In general we have a mixed integer problem (MIP1)

$$\min cx + dy$$

subject to

$$\begin{aligned} Ax + By &\geq b \\ Dx &\geq e \\ x &\in \mathbb{Z}^{n_1} \\ y &\in \mathbb{R}^{n_2} \end{aligned}$$

Let  $X = \{x \in \mathbb{Z}^{n_1} : Dx \geq e\}$  then MIP1 can be reformulated as:

$$\min_{\bar{x} \in X} \left\{ c\bar{x} + \min\{dy : By \geq b - A\bar{x}, y \in \mathbb{R}^{n_2}\} \right\} \quad (5.10)$$

The inner minimization is a linear program ( $\bar{x}$  are merely constants in this problem), which we denote the *primal Benders sub-problem* (PBSP). To ease the following we will assume that the inner minimization problem is feasible and

bounded for all choices of  $\bar{x} \in X$  since this is the case for our decomposition (as will be explained in the sequel). We note that in general Benders decomposition also applies when these assumptions do not hold, but is slightly more complex to handle, see for example Benders (1962) or Costa (2005) for details.

If we let  $\pi$  be the dual variables corresponding to  $By \geq b - A\bar{x}$  then we can write the dual of the inner minimization as:

$$\max\{\pi(b - A\bar{x}) : \pi B \leq d, \pi \geq 0\}.$$

This problem is denoted the *dual Benders sub-problem* (DBSP). Since we assumed  $\min\{dy : By \geq b - A\bar{x}, y \in \mathbb{R}^{n_2}\}$  to be feasible and bounded the PBSP will be feasible and bounded as well. Using the DBSP and strong duality we can rewrite (5.10) to:

$$\min_{\bar{x} \in X} \left\{ c\bar{x} + \max\{\pi(b - A\bar{x}) : \pi B \leq d, \pi \geq 0\} \right\} \quad (5.11)$$

Here we notice that the constraints of the inner maximization problem are independent on the choice of  $\bar{x} \in X$ . Furthermore,  $F = \{\pi B \leq d, \pi \geq 0\}$  is bounded and non-empty due to our assumptions and we can use Minkowski-Weyl's Theorem to express  $F$  using a set of extreme points  $\Pi = \{\pi_1, \dots, \pi_q\}$ . DBSP will have an optimal solution at one of the extreme points in  $\Pi$  and we can reformulate (5.11) to:

$$\min_{\bar{x} \in X} \left\{ c\bar{x} + \max\{\pi(b - A\bar{x}) : \pi \in \Pi\} \right\}$$

using an auxiliary variable  $z \in \mathbb{R}$  this problem can be written as:

$$\min c\bar{x} + z$$

subject to

$$z \geq \pi(b - A\bar{x}) \quad \pi \in \Pi \quad (5.12)$$

$$\bar{x} \in X \quad (5.13)$$

$$z \in \mathbb{R} \quad (5.14)$$

This problem is denoted the *Benders master problem* (BMP). The constraints (5.12) are known as *optimality cuts*. This BMP is usually solved in an iterative fashion since the cardinality of  $\Pi$  is such that enumerating all extreme points is out of question.

The lower bound on the optimal objective value is always a monotonic increasing function as it is obtained from the relaxed master problem where more and



more constraints, (extreme points), are added and the formulation continuously gets tighter. However, the upper bound or objective value is not guaranteed to be a monotonic decreasing function since it is just produced by a sequence of feasible solutions but by maintaining the best found solution the algorithm converges to the optimal solution.

We apply Benders decomposition to the speed optimization problem (5.1)-(5.9) such that  $(x_a, L_r)$  are found in the Benders master problem, BMP. This means that constraints (5.2), (5.5), (5.6), (5.8), and (5.9) are moved to the master problem while constraints (5.3), (5.4), and (5.7) are moved to the primal subproblem, which is given by:

$$\min \sum_{k \in K} \sum_{p \in P^k} r_p^k y_p^k \quad (5.15)$$

subject to

$$\sum_{p \in P^k} y_p^k \leq d^k \quad k \in K \quad (5.16)$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq \bar{x}_a q_a \quad a \in A \quad (5.17)$$

$$y_p^k \in \mathbb{R}_+ \quad k \in K, p \in P^k \quad (5.18)$$

where  $\bar{x}_a$  is the value of the  $x_a$  variables chosen in the master problem. It is intuitive to view constraint (5.17) as the two constraints (5.19) and (5.20)

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq q_a \quad a \in O(\bar{x}) \quad (5.19)$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq 0 \quad a \in C(\bar{x}) \quad (5.20)$$

where  $O(\bar{x})$  and  $C(\bar{x})$  denote “open” and “closed” arcs. The open arcs are the arcs with  $x_a = 1$  in the BMP and the closed arcs have  $x_a = 0$ . The PBSP is always feasible since setting all  $y_p^k$  equal to 0 produces a feasible solution. It is also bounded since constraints (5.16) and (5.18) ensure that  $0 \leq y_p^k \leq d^k$  for all  $a \in A, k \in K, p_k \in P^k$ .

The PBSP can be identified as the cargo routing multi-commodity flow problem, MCF, which can be solved efficiently using column generation. When iterating through solutions from the BMP, this method allows that some columns can be reused. We introduce side constraints on the transit time such that we solve a time-constrained multi-commodity flow problem and hence determine

an optimal speed selection taking transit time restrictions into consideration as done in Karsten et al. (2015).

To derive the optimality cuts for the BMP we associate with (5.16)-(5.20) the non-positive dual variables  $\alpha_k^i$ ,  $\delta_a^i$ , and  $\lambda_a^i$ . Then an extreme point solution gives a new Benders cut, which can be added to the BMP

$$z_0 \geq \sum_{k \in K} \alpha_k^i d_k + \sum_{(a) \in O(\bar{x})} \delta_a^i q_a x_a + \sum_{(a) \in C(\bar{x})} \lambda_a^i q_a x_a \quad (5.21)$$

With the set of all Benders cuts,  $BC$ , the BMP can be written as:

$$\min \sum_{a \in A} c_a x_a + \sum_{r \in R} C^r L^r + z_0 \quad (5.22)$$

subject to

$$z_0 \geq \sum_{k \in K} \alpha_k^i d_k + \sum_{(a) \in O(\bar{x})} \delta_a^i q_a x_a + \sum_{(a) \in C(\bar{x})} \lambda_a^i q_a x_a \quad i \in BC \quad (5.23)$$

$$\sum_{a \in A(\bar{a})} x_a = 1 \quad \bar{a} \in \bar{A} \quad (5.24)$$

$$\sum_{a \in E(r)} t_a x_a \leq 7L_r \quad r \in R \quad (5.25)$$

$$\sum_{r \in R(v)} L_r \leq N^v \quad v \in V \quad (5.26)$$

$$x_a \in \{0, 1\} \quad a \in A \quad (5.27)$$

$$L^r \in \mathbb{Z}_+ \quad r \in R \quad (5.28)$$

where the Benders cuts (5.23) are added iteratively.

### 5.4.1 Decomposition and Solution of the MCF Problem

The PBSP, (5.15)-(5.18), is the path-flow formulation of a multi-commodity flow problem. It has  $|A| + |K|$  constraints, but the number of variables (paths) grows exponentially with the size of the graph in the worst case. The necessary variables can be generated dynamically using another decomposition technique, namely column generation, and in practice the path-flow model can be solved efficiently even for very large scale instances, see Karsten et al. (2015). Column generation works with a reduced version of the LP (5.15)-(5.18) defined by a reduced set of columns  $\bar{P}^k$  for each commodity  $k$  such that a feasible solution can be found using variables from  $\cup_{k \in K} \bar{P}^k$ . Solving this LP gives rise to dual

variables  $\alpha_k$  and  $\delta_a$  corresponding to constraint (5.16) and (5.17), respectively. For a path variable  $j \in P$  let  $\kappa(j)$  denote the commodity that a variable serves,  $p(j)$  the path (set of arcs) corresponding to the variable  $j$ , and  $c_a^{\kappa(j)}$  the cost of sending one unit through arc  $a$ . The *reduced cost*  $\bar{c}_j$  of each path variable  $j \in P$  is  $\bar{c}_j = \sum_{a \in p(j)} (c_a^{\kappa(j)} - \delta_a) - \alpha_{\kappa(j)}$  and we wish to find variables such that  $\bar{c}_j < 0$ , as this variable can potentially improve the current LP solution and give new dual variables. To find a variable with negative reduced cost or prove that no such variable exists, we solve a shortest path problem for each commodity from the source to the destination on the reduced cost graph. As we want to accommodate the transit time restrictions for each commodity, we use a resource constrained shortest path algorithm with time as the resource to ensure that the transit time of each generated path is less than or equal to the maximum transit time for the given commodity as described in Karsten et al. (2015). Transit time is in addition to the sailing legs calculated by considering the multi-commodity flow problem on a graph including transshipment, loading, and unloading arcs.

We can add Benders cuts based on the LP-relaxation of the BMP as the right hand side of (5.17) is multiplied by the capacity  $q_a$  of each arc,  $a \in A$ , such that this will correspond to solving the same time constrained MCF problem but on a multi graph where the “fractional” capacity of parallel arcs will sum to the original capacity.

In both cases we can warm start the column generation procedure by using the columns from previous configurations.

### 5.4.2 Strengthening the Benders Cuts

From duality we can gain some additional insights on the dual values associated with the “closed” arcs. Let  $p(O(\bar{x}))$  and  $p(C(\bar{x}))$  be the set of “open” and “closed” arcs used by path  $p$ . The DBSP is:

$$\max \sum_{k \in K} \alpha_k + \sum_{a \in O(\bar{x})} q_a \delta_a + \sum_{a \in C(\bar{x})} 0 \lambda_a \quad (5.29)$$

subject to

$$\alpha_k + \sum_{a \in p(O(\bar{x}))} \delta_a + \sum_{a \in p(C(\bar{x}))} \lambda_a \leq r_p^k \quad k \in K, p \in P^k \quad (5.30)$$

$$\alpha_k, \delta_a, \lambda_a \leq 0 \quad k \in K, a \in A \quad (5.31)$$

Since we want the Benders cut to be as strong as possible, we can optimize  $\lambda_a$  in the dual as it does not contribute to the objective value. For a given solution where we obtain the dual values  $\alpha_k^*$  and  $\delta_a^*$  we want to find an alternative solution where  $\alpha_k$  and  $\delta_a$  take the values of  $\alpha_k^*$  and  $\delta_a^*$  but where  $\sum_{a \in C} \lambda_a \geq \sum_{a \in C} \lambda_a^*$ . This can be done by solving the following problem:

$$\max \sum_{a \in C(\bar{x})} \lambda_a \quad (5.32)$$

subject to

$$\sum_{a \in p(C(\bar{x}))} \lambda_a \leq r_p^k - \alpha_k^* - \sum_{a \in p(O(\bar{x}))} \delta_a^* \quad k \in K, p \in P^k \quad (5.33)$$

$$\lambda_a \leq 0 \quad a \in C \quad (5.34)$$

Let  $y_p^k$  be the dual corresponding to constraint (5.33), then we get the dual problem (corresponding to the PBSP):

$$\min \sum_{k \in K} \sum_{p \in P^k} \left( (r_p^k - \sum_{a \in p(O(\bar{x}))} \delta_a^*) - \alpha_k^* \right) y_p^k \quad (5.35)$$

subject to

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq 1 \quad a \in C(\bar{x}) \quad (5.36)$$

$$y_p^k \geq 0 \quad k \in K, p \in P^k \quad (5.37)$$

The problem has a similar structure to the original problem and can be solved using column generation as well. For the set of columns,  $\Delta^k$ , the reduced cost for a path variable  $l \in \cup_{k \in K} \Delta^k$  with original revenue/cost  $r_p^k - \alpha_k^* - \sum_{a \in p(O(\bar{x}))} \delta_a^*$  is given by the following resource constrained shortest path problem:

$$\bar{c}_l = \sum_{(a) \in p(l)} (c_a - \delta_a^* - \lambda_a) - \alpha_{k(l)}^* \quad (5.38)$$

The columns for a given commodity are added to the master problem when the reduced cost is less than the revenue associated with the commodity. When the solution to (5.32)-(5.34) is different from the initially found duals we can add an additional Benders cut. However, this cut does not necessarily dominate the original cut. Again we can reuse all columns corresponding to paths using at least one closed arc to warm start the column generation.

### 5.4.3 Generating Additional Benders Cuts

Generally, the Benders decomposition approach is more successful for standard multi commodity network design problems when the BSP decomposes into even smaller sub-problems. It can decompose e.g. by commodity (Gendron, 2011) or by equipment type (Cordeau et al., 2000) and especially if these can be solved by special purpose algorithms (Magnanti and Wong, 1981) such that more cuts can be added very effectively in each iteration. In the present problem the multi-commodity flow problem is not separable by commodity or equipment type but it is still possible to generate several alternative cuts in each iteration. In Appendix 5.8.2 we describe a method for generating cuts in other areas of the solution space based on the solution found to the multi-commodity flow problem.

### 5.4.4 Valid Inequalities

In this section we consider valid inequalities for the Benders master problem. We will solely focus on inequalities defined on the  $x_a$  and  $L_r$  variables, thus omitting the  $z$  variable.

We first introduce a lower limit on the number of needed vessels at a rotation by looking at the maximum speed for each arc in a rotation.

$$L_r \geq y_{min}^r = \left\lceil \sum_{\bar{a} \in \bar{A}(r)} \min_{a \in A(\bar{a})} t_a / 7 \right\rceil \quad r \in R. \quad (5.39)$$

We add (5.39) to the BMP to strengthen the model. We can also define

$$L_r \leq y_{max}^r = \left\lceil \sum_{\bar{a} \in \bar{A}(r)} \max_{a \in A(\bar{a})} t_a / 7 \right\rceil \quad r \in R \quad (5.40)$$

which will not cut away any optimal solutions, but is strictly speaking not a valid inequality.

Next, we consider valid inequalities that can be constructed based on the frequency constraints (5.25) along with the domain definitions for the variables and the arc selection constraint (5.24). In other words, for each  $r \in R$  we are

interested in the set:

$$\mathcal{B}_r = \left\{ x_a \in \{0, 1\} \quad \forall a \in E(r), L_r \in \mathbb{Z}_+, \right. \\ \left. \sum_{a \in E(r)} t_a x_a \leq 7L_r, \sum_{a \in A(\bar{a})} x_a = 1, \quad y_{\min} \leq L_r \leq y_{\max} \right\}$$

and valid inequalities for the polyhedron:

$$F_r = \text{conv}\{\mathcal{B}_r\}.$$

Some families of valid inequality for a similar polyhedron (without the arc selection constraint) have been proposed in Atamtürk and Rajan (2002). For the instances we are considering it is, in practice, relatively easy to optimize over  $F_r$ , and we have observed that the arc selection constraint improves the performance compared to not including it. Therefore, we will attempt to find any possible valid inequality for the polyhedron using a cut-finding LP. The use of cut-finding LPs to find all violated valid inequalities for a given polyhedron has, for example, been used by Boyd (1994), Boccia et al. (2008) and Kaparis and Letchford (2010).

#### 5.4.4.1 Separation

For a given solution to the relaxation of the master problem,  $(x_a^*, L_r^*)$ , we wish to determine a valid inequality  $\sum_{a \in E(r)} \pi_a x_a + \pi_r L_r \leq \beta_r$  for  $F_r$  that is violated by the solution. We are to determine the values of  $\pi_a$ ,  $\pi_r$  and  $\beta_r$  such that the inequality is valid and violated by  $(x_a^*, L_r^*)$ . We say that  $\sum_{a \in E(r)} \pi_a x_a^* + \pi_r L_r^* - \beta_r$  is the *violation* of the inequality. This can be done by solving the following LP:

$$\max \sum_{a \in E(r)} \pi_a x_a^* + \pi_r L_r^* - \beta_r \quad (5.41)$$

subject to

$$\pi_a \tilde{x}_a + \pi_r \tilde{L}_r \leq \beta_r \quad (\tilde{x}_a, \tilde{L}_r) \in \mathcal{B}_r \quad (5.42)$$

$$-1 \leq \pi_a \leq 1 \quad a \in E(r) \quad (5.43)$$

$$-1 \leq \pi_r \leq 1 \quad (5.44)$$

$$\pi_a, \pi_r, \beta_r \in \mathbb{R} \quad a \in E(r) \quad (5.45)$$

The objective function (5.41) maximizes the violation of the inequality. The first constraints (5.42) ensure that the inequality is satisfied by all solutions

from  $\mathcal{B}_r$  and therefore is a valid inequality and constraints (5.43)-(5.44) normalizes the inequality. Without these constraints, the LP would be unbounded whenever a violated inequality exists (since such a constraint can be scaled to yield any violation). Given that the  $\pi_a$  and  $\pi_r$  are now bounded we can further limit the range of  $\beta_r$ :

$$-|E(r)| - y_{max}^r \leq \beta_r \leq |E(r)| + y_{max}^r$$

Since the set  $\mathcal{B}_r$  can be prohibitively large, we initially remove the constraints (5.42) and add them dynamically when violated. Given a solution  $(\pi_a^*, \pi_r^*, \beta_r^*)$  to the partial cut-finding LP (5.41), (5.43)-(5.45) and a subset of constraints (5.42) the separation problem for constraints (5.42) is:

$$\max \left\{ \sum_{a \in E(r)} \pi_a^* x_a + \pi_r^* L_r - \beta_r^* : (x_a, \beta_r) \in \mathcal{B}_r \right\}$$

which written in full is:

$$\max \sum_{a \in E(r)} \pi_a^* x_a + \pi_r^* L_r - \beta_r^* \tag{5.46}$$

subject to

$$\sum_{a \in E(r)} t_a x_a \leq 7L_r \tag{5.47}$$

$$\sum_{a \in A(\bar{a})} x_a = 1 \quad \bar{a} \in \bar{A} \tag{5.48}$$

$$y_{min}^r \leq L_r \leq y_{max}^r \tag{5.49}$$

$$x_a \in \{0, 1\} \quad a \in E(r) \tag{5.50}$$

$$L_r \in \mathbb{Z}_+ \tag{5.51}$$

If this IP has a positive value function then we have detected a solution  $(\tilde{x}_a, \tilde{L}_r)$  in  $\mathcal{B}_r$  that is violated by the inequality given by  $(\pi_a^*, \pi_r^*, \beta_r^*)$  and we add

$$\pi_a \tilde{x}_a + \pi_r \tilde{L}_r \leq \beta_r$$

to the cut finding LP and resolve.

## 5.5 Algorithm

Traditional implementations of the Benders decomposition algorithm follow a cutting plane approach where the reduced master problem is solved iteratively to optimality, and a constraint of type (5.23) (Benders cut) is added in each iteration based on the sub-problem. This procedure is followed iteratively until optimality is reached (or the bounds are within an acceptable tolerance). This has the downside that too much time may be spent on proving optimality and re-processing nodes of the branch-and-bound tree that have already been cut off every time a new Benders cut is added. However, most modern branch-and-bound solvers make it possible to effectively make branch-and-cut algorithms where cuts are added using a callback routine as described by Bai and Rubin (2009) and Fortz and Poss (2009). Using callbacks makes it possible to add cuts efficiently both at integer and LP solutions whenever one is found. This comes at the cost of potentially adding too many cuts, but a node will never have to be revisited. Both implementations have advantages for different types of problems, which we will discuss in the computational section. To improve the implementation of the Benders algorithm, we also test the effect of solving the LP-relaxation of the BMP (we relax the variables related to the number of vessels on a rotation (5.28) and arc selection variables (5.27)). After solving the LP-relaxation of the BMP, we add the Benders cuts obtained from this to an initial pool of cuts before eventually solving the integral version of the problem. This procedure has been shown to be very effective by e.g. Cordeau et al. (2001) and Fortz and Poss (2009). Additionally, we add one warm starting cut a priori based on a known initial configuration of the network, which is usually quite good and hence can be expected to improve performance. We terminate the algorithm when a relative gap of 1 % between the best found solution and the lower bound is achieved, and set the tolerance of the mixed integer programming solver to 1 % as well. The valid inequalities described in Section 5.4.4 based on the frequency constraints are added dynamically as cuts using callbacks in both the traditional and branch-and-cut approach. They are added to the LP-relaxation as well as the BMP, but only at the root node. The inequalities (5.39) are always added a priori. The column generation procedure for solving the multi-commodity flow problem is re-using previously generated columns to warm start the algorithm for each new configuration of the network, but to manage the number of columns, unused columns are deleted every 100<sup>th</sup> iteration of the overall algorithm. Additionally, we keep columns generated for the initial configuration. In the column generation procedure used to generate the strengthened Benders cuts described in Section 5.4.2, we only reuse columns containing closed arcs to warm start the procedure.

The model is implemented in C++. We use the Boost Graph Library to handle



Name	$ R $	$ K $	$ \bar{A} $	$ A $
<b>Baltic</b>	3	22	14	46
<b>WAF</b>	10	37	40	130
<b>Mediterranean</b>	5	365	58	189
<b>Pacific</b>	14	722	141	464
<b>WorldSmall</b>	26	1,764	287	951

**Table 5.1:** Characteristics of the considered networks.  $|R|$  is the number of rotations,  $|K|$  is the number of commodities,  $|\bar{A}|$  is the number of original sailing arcs, and  $|A|$  is the number of potential sailing arcs. In addition to the sailing arcs our formulation adds (un)load and transshipment arcs.

the graph construction. The BMP is solved using Gurobi 6.0 and the PBSP using the COIN-OR linear programming solver. All tests were performed using a single thread on a computer with an Intel Xeon CPU X5550 2.67GHz. We allow the algorithm to run for up to three hours and if the LP-relaxation is solved initially, up to one of the three hours is dedicated to this.

## 5.6 Computational Results

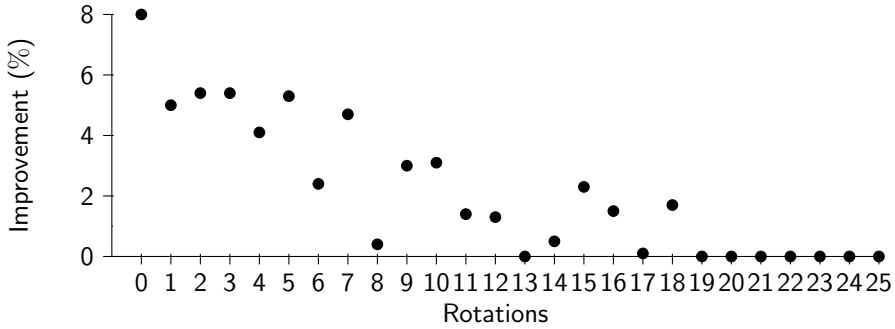
We test the algorithm as a post-processing tool on networks created based on realistic data from *Linerlib* (Brouer et al., 2014a) using the matheuristic described in Brouer et al. (2014b, 2015). A summary of the considered networks can be seen in Table 5.1. We use the average speed configuration for generating the warm starting cut, whereas in a real world network we would have an already optimized configuration which often could give a good quality cut. If there are no transit time sensitive demands being transported by a given rotation, the most cost effective configuration will be sailing all legs with average speed while maintaining a weekly frequency.

We create multigraphs by considering up to five possible sailing arcs for each original sailing arc in the rotation(s) being optimized such that the input speed, as well as duplicates corresponding to  $\pm 10\%$  and  $\pm 25\%$  speed change, are considered and added if the resulting speed is feasible for the vessel class. For the Panamax 2400 vessel class sailing at 17 nm/h on a specific arc this corresponds to arcs at 12.75, 15.3, 17, 18.7, and 21.25 since these are all within the speed limits of 12 and 22 nm/h. We set the loading and unloading time to one day and the transshipment time to two days as in Brouer et al. (2015).

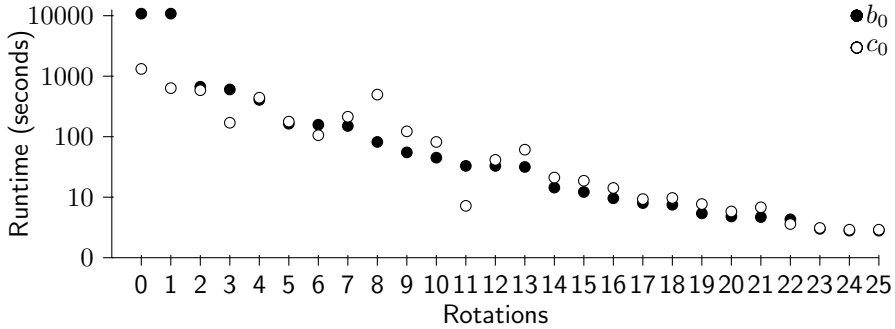
We consider optimization at three levels. Each rotation can be optimized separately (while still considering the cargo routing in the entire network), all rotations using the same vessel class can be optimized jointly, and finally all the rotations in a network can be optimized simultaneously. Additionally, rotations can be optimized such that the number of vessels used by each rotation is maintained i.e.  $L_r$  is *fixed* at the current deployment or the number of vessels used by each rotation can be optimized as well such that the fleet deployment for each rotation,  $L_r$ , is *flexible* within the bounds given by the total available fleet. Optimizing for a fixed fleet will be more relevant under operational planning whereas the flexible fleet optimization is more targeted at tactical planning. In both cases the cargo routing in the entire network is considered. In the following we consider optimization at the three levels for a fixed and flexible fleet. Additionally, we will discuss the impact of the different proposed algorithmic improvements for the vessel class and network optimization. In the vessel class optimization we present averaged results over several classes, but when we find it relevant we also refer to the underlying detailed results, which are not shown to keep the size of the tables manageable.

### 5.6.1 Single Rotation Optimization

Figure 5.4 shows the improvement for each rotation in the WorldSmall instance where the fleet is flexible. The average improvement over all rotations is 2 % and the maximum improvement is 8 % of the total profit. For most rotations the total cargo routed is unchanged or slightly reduced (less than 0.2 % change in total volume) and for three of the rotations the speed optimization leads to a slight increase in volume of the cargo routed. For all rotations the deployment is either the same or increased i.e. the overall average speed is decreased. This illustrates that in this case the improvements in profit are mainly driven by an overall speed decrease (some sailing legs maintain or increase speed) and change in deployment, but in a few cases also by an increase in the volume of cargo routed. Looking at the container paths used for the cargo routing in the optimized solution (not just different speed but new itinerary or different rotation between same ports) reveals that it is essential to consider cargo routing as part of the speed optimization. On average 9 % of the container paths used in the best found solution are not used by the initial solution. The variation is between 0 % and 20 %. Some of the difference may be accounted for by parallel paths having the same cost and itinerary but using different rotations. Hence, from an objective function point of view they are equally good, but lead to different routings. In practice it may be desired to have as much of the routing unchanged as possible when optimizing speed, and simple modifications that give preference to unchanged flow can easily be incorporated in



**Figure 5.4:** Percentage improvement in profit in the WorldSmall instance when optimizing each rotation. The improvements are sorted according to the runtime for the branch-and-cut implementation without any improvements.



**Figure 5.5:** Runtime for each rotation in the WorldSmall instance sorted according to the runtime for the traditional Benders implementation without any improvements  $b_0$ .  $c_0$  is the branch-and-cut implementation without any improvements. Notice the logarithmic scale of the y-axis.

the multi-commodity flow problem.

Figure 5.5 shows the runtime for each rotation in the WorldSmall instance for the traditional and branch-and-cut implementation of the Benders algorithm without any algorithmic improvements. The runtimes are sorted according to the traditional implementation. For the rotations that take longer time to optimize, the branch-and-cut implementation is generally faster than the traditional implementation, and it also solves all instances within the time limit, whereas the gap is not closed for two of the rotations using the traditional approach. For the rotations where the optimal solution is found in short time, the traditional implementation converges to the desired solution quality slightly faster.

### 5.6.2 Vessel Class Optimization

When the number of vessels is restricted, all configurations of the network may not be possible and e.g. an overall speed decrease of all rotations may use more vessels than are available. Hence, it is desirable to optimize rotations with the same class deployed simultaneously rather than each rotation individually, as it leads to overly optimistic profit improvements corresponding to infeasible deployments. In the following we show results for the vessel class optimization in the WorldSmall instance where a major decision in the speed optimization process is the deployment of vessels to rotations. The results can be found in Table 5.2 and some details on solution characteristics for each vessel class (F450, F800, P1200, P2400, PostP, and SuperP where F is feeder and P is Panamax) in the flexible deployment case can be found in Table 5.3.

#### Fixed Deployment

As seen in Table 5.2, the potential improvements in profit are less than 1 % on average for all classes when the fleet is fixed and speed changes of 10 % and 25 % on each sailing leg are allowed. However, even 0.2 % is significant for a global liner shipping network. The detailed results show that in all cases the volume of containers routed is either maintained or increased slightly (less than 1 % increase in volume transported). Generally, the solution times are low but the branch-and-cut implementation is superior. The traditional implementation only solves the problem to within the desired tolerance within the time limit for all six vessel classes when Benders cuts based on the LP-relaxation and the warm start cut is added initially. In the branch-and-cut implementation all six classes are solved when the LP-relaxation is solved initially and a warm start cut is added. The best average solution time, where all six classes were solved, is achieved when valid inequalities are also added to the LP-relaxation as well as at the root node of the integral BMP, and the algorithmic improvements focusing on the bound does not hurt performance as all six instances are solved. For the five instances that are solved by all configurations, the pure branch-and-cut implementation without any improvements is superior.

#### Flexible Deployment

Table 5.2 also shows results for the flexible fleet case and Table 5.3 shows the solution characteristics for the best found solution for each class. For three

Average over all vessel classes in WorldSmall																				
Traditional implementation							Branch-and-cut implementation													
LP-relaxation cuts							Warm start cut													
Strengthened cuts							Node relaxation cuts													
Valid inequalities							Improvement in profit (%)													
							Final gap (FUB-FLB)/ FUB  (%)													
							Final UB gap (FUB*-FUB)/ FUB*  (%)													
							Final LB gap (FUB*-FLB)/ FUB*  (%)													
							Initial LB gap (FUB*-ILB)/ FUB*  (%)													
							Runtime for solved classes (seconds)													
							Solved classes (out of 6)													
							Improvement in profit (%)													
							Final gap (FUB-FLB)/ FUB  (%)													
							Final UB gap (FUB*-FUB)/ FUB*  (%)													
							Final LB gap (FUB*-FLB)/ FUB*  (%)													
							Initial LB gap (FUB*-ILB)/ FUB*  (%)													
							Runtime for solved classes (seconds)													
							Solved classes (out of 6)													
Setting							Fixed deployment							Flexible deployment						
×							0.2	4.0	0.2	3.8	-	23	5	4.8	34.5	1.3	32.4	-	23	3
×		×					0.1	0.2	0.3	0.2	0.7	336	6	3.2	33.7	5.3	25.1	27.6	27	3
×		×	×				0.1	0.5	0.3	0.2	0.9	1,197	6	3.2	21.6	2.7	17.7	20	23	3
×		×	×	×			0.1	0.5	0.3	0.2	0.9	22	5	1.6	23.7	4.1	17.7	20	27	3
×		×	×			×	0.1	2.4	0.3	2.2	2.8	26	5	2.7	52.7	4.5	43.6	45	31	3
	×						0.2	2.6	0.1	2.4	-	7	5	3.3	51.1	1.7	47.0	-	12	3
	×	×					0.2	0.6	0.2	0.4	0.4	20	5	3.3	11.1	1.7	9.0	9.5	47	3
	×	×	×				0.2	0.5	0.1	0.4	0.4	455	6	3.0	11.7	2.0	9.1	9.6	42	3
	×	×	×	×			0.2	0.5	0.1	0.4	0.4	651	6	3.0	11.6	1.9	9.1	9.6	79	3
	×	×	×		×		0.2	0.5	0.1	0.4	0.4	427	6	2.2	12.4	2.7	8.9	9.6	53	3
	×	×	×			×	0.2	0.5	0.3	0.4	0.4	283	6	4.3	9.6	0.8	8.6	8.7	71	3

Vessel class (size in FFE)	Number of rotations	Flow (chg. in %)	Additional vessels deployed	Profit (chg. in %)	Final gap (%)
F450	3	0	0	0	<1
F800	3	0	0	0	<1
P1200	5	-0.3	11	8	6
P2400	5	-0.6	11	13	27
PostP	7	-0.6	6	10	19
SuperP	1	0	0	0	<1

**Table 5.3:** Solution characteristics of the best found solution for each vessel class in the WorldSmall instance with flexible vessel deployment. We report the flow as the change in volume of the container routing, the number of additional vessels deployed, the change in overall profit for the network, and the final gap for the best found solution.

the final lower bound is generally poor. Improvements up to 12.8 % are found and it is clear that most of the improvements are due to changes in deployment such that the overall average sailing speed is lowered. For rotations assigned the vessel classes F450, F800, and SuperP no improvements larger than 1 % can be made, and the detailed computational results for these (not shown) show that the optimization terminates quickly. This characteristic was also found in the single rotation optimization case. For the three remaining vessel classes, P1200, P2400, and PostP significant improvements in the profit can be made. Inspection of the solutions also reveal that the sailing speed on some sailing legs are maintained or increased to meet critical transit times for some demands, and as seen only a few demands cannot be met. For all classes when deployment is flexible the volume of containers routed is either maintained or slightly decreased (less than 0.6 %). Further improvements may be achievable for all classes if different/more sailing speeds are considered. For the best found solution the weekly fuel cost is reduced from \$ 68 mio. to \$ 64 mio., the weekly time charter rate of 11 additional vessels is \$ 1.6 mio. and cargo revenue decreases from \$ 132.2 mio to \$ 131.7 mio.

If we look at all rotations in the WorldSmall instance using the P1200 vessel class we see a significant improvement in profit and the smallest gap of the classes leading to an improved solution. Here we use 11 extra vessels which is exactly what is available in the instance. If we sum the profit for the results of the individual rotations from Section 5.6.1, which are all solved to the desired tolerance, a total improvement in profit of 9 % is possible, but it also requires 12 additional vessels, which are not available. For the PostP vessel class seven additional vessels are available, but the optimal solution to the optimization of the individual rotations uses 12 additional vessels in this case. For the P2400

vessel class the best found solution uses the same number of vessels as the optimal solution for each of the individual rotations, so in this case the single rotation optimization could provide a very good warm start solution (but not optimal as transit time critical containers may use two linked rotations from the same class where joint optimization could lead to a lower selected speed on both). As expected there is a larger change in container paths used for the cargo routing when optimizing all rotations in a class rather than a single rotation. On average 11 % of the container paths used in the solutions are not used by the initial solution and the variation is between 0 % and 40 %. The variation is correlated with how much the network is improved, but the largest improvements do not necessarily lead to the largest changes in the container routing.

Adding valid inequalities and the solution of the LP-relaxation initially improves the performance of the algorithm in terms of improving both the lower and upper bound. Adding the strengthened Benders cuts and Benders cuts at node relaxations generally do not improve performance in terms of best solution for vessel class optimization given the time limit, but does improve the bound. The number of basic Benders cuts added by the algorithm (not reported) is generally lower when additional/strengthened cuts are added as one “iteration” takes longer time. (We have implemented the additional cuts discussed in Appendix 5.8.2 and also here the number of iterations is reduced, but it generally neither improves or deteriorates performance significantly). This means that the derived cuts including the ones discussed in Appendix 5.8.2 do improve the “per iteration” performance, and for the instances that are solved the number of added basic Benders cuts is lower. When we are interested in solutions quickly (or better solutions with less guarantees on quality) it can be advantageous to use a more basic implementation to reduce the time spent on improving the LB. On the other hand the convergence of the LB is very slow when no improvements are made and if the gap is too large only poor solutions may be found. The initial lower bound gap is only reported when the LP-relaxation of the problems is solved initially (otherwise a dash) and here the branch-and-cut implementation usually has better progress, and the detailed results show that more Benders cuts are added within the time limit. When the valid inequalities are added in the traditional implementation, the progress on the LP is very slow whereas they improve performance in the branch-and-cut implementation. Warm starting the algorithm using a known configuration helps to ensure a good initial solution and possible improvements can quickly be assessed.

### 5.6.3 Network Optimization

The majority of cargoes in real liner shipping networks uses up to one transshipment to travel from origin to destination port. Still, a significant amount of cargo uses two or more transshipments. Hence, speed optimization decisions across several rotations, potentially with different capacity, influence the cargo routing. The proposed method allows optimization of an entire network or parts of a network e.g. within a region or some other grouping of rotations based on current container routing. It should be noted that in practice all rotations in a global network are usually not optimized simultaneously.

Table 5.4 shows the results for optimization of networks where all rotations are optimized simultaneously. We show results for two algorithmic settings. Setting  $s_1$  is the branch-and-cut implementation with a warm start cut and solution of the LP initially. Setting  $s_2$  is as  $s_1$ , but we also add the strengthened Benders cuts, Benders cuts at node relaxations and valid inequalities at the LP and the root node. Generally, setting  $s_1$  is faster on small instances and for the larger instances setting  $s_2$  improves the initial and final gap significantly. For instances covering the Baltic and WAF, the algorithm performs very well and finds optimal solutions quickly (showing that no improvement can be made with fixed deployment) in both setting  $s_1$  and  $s_2$ . For all instances it is seen that valid inequalities significantly improve the initial gap in setting  $s_2$ . In both Pacific and WorldSmall there are issues with convergence of the LP and a significant final gap is reported. In WorldSmall no improvement is found within the time limit, but we know from the vessel class optimization that a significant improvement is achievable. However, in the flexible case, setting  $s_2$  provides a better initial and final gap whereas setting  $s_1$  is better in the more constrained fixed case. If the time limit is increased by a factor 4 such that we allow 12 hours in total and up to 4 hours at solving the LP, setting  $s_2$  is able to improve the network by 5.2 % whereas setting  $s_1$  still is not. Furthermore, we see that solving the LP initially is very effective in improving the bound (but still it does not converge within the time limit), whereas less improvement is achieved in the integer phase. Generally, for the larger instances setting  $s_1$  shows better performance when the fleet is fixed, and setting  $s_2$  shows better performance when the fleet is flexible.

The network results show that the method in its current form is not suitable for optimization of an entire network as it has problems with convergence for the largest instances. However, for simultaneous optimization of all rotations in a network the algorithmic performance can potentially be improved by solving the problem in several stages. Initially the single rotation optimization can serve as input to vessel class optimization, which can eventually serve as input to optimization of the full network. To find a feasible solution that can serve



Network															
Branch-and-cut implementation	LP-relaxation cuts	Warm start cut	Strengthened cuts	Node relaxation cuts	Valid + inequalities		Improvement in profit (%)	Final gap (FUB-FLB)/ FUB  (%)	Initial LB gap (FUB*-LPLB)/ FUB*  (%)	Runtime (seconds)		Improvement in profit (%)	Final gap (FUB-FLB)/ FUB  (%)	Initial LB gap (FUB*-LPLB)/ FUB*  (%)	Runtime (seconds)
Algorithmic setting						Fixed deployment					Flexible deployment				
Baltic															
x	x	x					0	0	0	0.01		0	0	0	0.01
x	x	x	x	x	x		0	0	0	0.01		0	0	0	0.01
WAF															
x	x	x					0	0.2	0.2	0.2		1.9	0.9	4.2	0.5
x	x	x	x	x	x		0	0.0	0.0	0.5		1.9	0.7	1.9	4.5
Mediterranean															
x	x	x					2.0	1.1	2.4	t.l.		2.0	1.2	1.8	t.l.
x	x	x	x	x	x		1.9	1.2	1.7	t.l.		2.0	1.2	1.6	t.l.
Pacific															
x	x	x					2.1	2.8	4	t.l.		3.1	48	108	t.l.
x	x	x	x	x	x		1.9	3.2	3.8	t.l.		7.9	70	73	t.l.
WorldSmall															
x	x	x					0.7	3.3	3.6	t.l.		0.0	374	375	t.l.
x	x	x	x	x	x		0.4	5.1	6.1	t.l.		0.0	365	349	t.l.
x	x	x					1.6	3.6	1.7	12h		0.0	265	262	12h
x	x	x	x	x	x		0.4	6.3	4.4	12h		5.2	275	256	12h

**Table 5.4:** Computational Results: The algorithm is tested on networks where all rotations are open to optimization. Improvement in profit is the improvement in profit obtained relative to the average speed configuration. Runtime is the total time to converge to within the desired tolerance including the time used at solving the LP and "t.l." indicates that the time limit of three hours has been reached. LPLB is the lower bound obtained from solving the LP and FUB is the final upper bound. The last row for WorldSmall show results where the time limit has been increased by a factor four and up to 4 hours can be spent on solving the LP.

as input for the class optimization, the rotations can be ranked according to expected impact and fleet usage can be updated after optimization of each rotation. The solution found in the class optimization will always be a feasible solution for optimizing the entire network.

## 5.7 Conclusions

In this paper we introduced a novel model and solution method for liner shipping companies to optimally determine the sailing speed of one or several rotations simultaneously in a global network. In the model and solution method we consider level of service explicitly, and incorporate cargo routing decisions with tight transit time restrictions on each commodity in the entire network. Furthermore, we show in Appendix 5.8.1 that it is possible to extend the model to include optimal time scheduling decisions in the model. The solution method is based on Benders decomposition and column generation and we show that it is able to effectively improve the profit of global size networks. We have used a state of the art algorithm to generate the networks used for testing, and our results show that variable speed on each sailing leg and fleet deployment can lead to large savings in the network design process. Also, we have shown that speed changes can lead to significant changes in the routing of the containers in a global network, and hence it is critical to consider routing implications when optimizing speed in networks where transit time restrictions are tight. In addition to the model and solution method we have proposed several algorithmic enhancements, which all helps improving the bound on the solution. The model and algorithm can be used as a basis for the development of decision support tools in liner shipping companies and it applies in both tactical and operational settings. We believe that even if extended with the time scheduling component described in Appendix 5.8.1 smaller initial gaps may be expected in an actual planning setting since a good configuration is already in place. Often only speed changes will be considered for smaller parts of the networks and potentially close to the existing operation in terms of speed variation. The model and solution method can handle the large and complex planning problems faced by leading liner shipping companies as networks are usually not entirely changed, but merely incrementally improved to new operating conditions. Therefore, future work could be in the direction of solving more restricted problems to improve the upper bounds and obtain solutions faster. Additionally, this can help improve the lower bounds. Finally, searching for improving solutions in the proximity (Fischetti and Monaci, 2014) of the best known solutions rather than based on relaxations may improve the performance.

## Acknowledgements

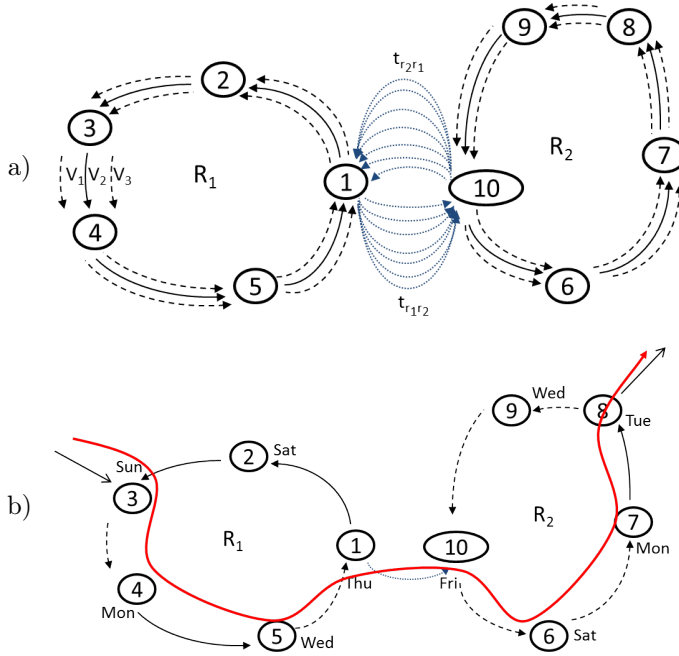
This project was supported by The Danish Maritime Fund under the Competitive Liner Shipping Network Design project and by the Danish Strategic Research Council and EUDP under the GREENSHIP project. We wish to thank managers and planners at Maersk Line for providing insights and data on their current network operations.

## 5.8 Appendix

### 5.8.1 Optimal Time Scheduling

It is possible to add time scheduling constraints to the model to include determination of an optimal time schedule while satisfying the transit time restrictions. The time schedule determines the timing of events, such as port calls, along a service. For each port we introduce a variable to determine the departure time, and at the same time we introduce variables to reflect transshipment time between rotations. These are coupled with the flow variables such that it is possible to consider the influence of schedules in the flow calculations. The time scheduling and transshipment decisions are included in the BMP. Duplicates of the transshipment arcs are considered with different layover time corresponding to all possible schedules, and a binary variable is associated with each of the transshipment arcs. The selected transshipment arc is included in the PBSP. If the departure time for some or all ports are given a priori, this is easy to include by fixing part of the network.

Figure 5.6 is an extension of the example in Figure 5.3 and Figure 5.6 a) shows the transportation network before the schedule and speeds have been selected. The transshipment time from rotation  $R_1$  to  $R_2$  will depend on the schedule of the two rotations, but since both schedule and speed is variable we need to consider different transshipment arcs (blue dotted lines in the figure). In this case there are seven arcs corresponding to a transshipment time of one to seven days. Figure b) shows an example flow from 3 to 8 where the schedule and speeds have been selected. Rotation  $R_1$  calls port 1 every Thursday and  $R_2$  departs from the same port (represented by a different port call) every Friday, so the transshipment arc used in this case is the one corresponding to one day. This way it is possible to calculate the transit time from 3 to 8 as the length of each sailing leg divided by the selected speed for each of the legs plus



**Figure 5.6:** Figure a) a transportation network used for speed and schedule optimization (the problem considered in the BMP). Figure b) the flow through a network at a specific speed and schedule (the problem considered in the PBSP). The nodes correspond to port calls, which are numbered consecutively around each service. The ten port calls correspond to nine physical ports (the port calls 1 and 10 correspond to the same physical port) and it is possible to transship between rotation  $R_1$  and  $R_2$  by using a transshipment arc from 1 to 10.

the transshipment time corresponding to the selected schedule. If the total transit time from the physical port corresponding to 3 to the physical port corresponding to 8 is longer than what is allowed for a commodity going on this path, there is no feasible path and it may be worth adjusting the speeds and the schedule.

### 5.8.1.1 Modeling

We wish to determine when each port is visited by which rotation and how this influences the achievable transshipment times and thereby the cargo routing. For each rotation the schedule (i.e. arrival and departure times) is determined in all ports for all rotations. The time it takes to transship between two different rotations visiting the same port is determined by the arrival in the port for each of the rotations and some buffer time. In the following we assume for simplicity weekly rotations, but the model can be extended to accommodate bi-weekly frequencies. The set of all port calls is  $I$  and in the example of nine physical ports in Figure 5.6 this corresponds to the port calls 1 to 10 which are consecutively numbered along each service. For each rotation  $r \in R$  we assign a starting port call  $\sigma_r$  a priori (port call 1 and 6 in Figure 5.6) and use this as reference for the schedule of the subsequent ports in each of the rotations. The starting port will always be the call with the lowest index within the rotation. Each service consist of several port calls, and in the case of butterfly rotations the same port may have multiple corresponding port calls. The set of physical ports is  $Q$  and the set of port calls in port  $q \in Q$  is  $I(q)$ . The arrival time at port call  $i \in I$  given in days is determined by the continuous decision variable  $T_i \in \mathbb{R}$  and the integer decision variable  $w_{i'i''} \in \mathbb{Z}$  is the offset in weeks between two port calls for the same port,  $i'$  and  $i''$ , i.e.,  $w_{i'i''}$  is not necessarily equal to  $w_{i''i'}$ . Usually two port calls  $i'$  and  $i''$  at port  $p$  correspond to two different rotations, but for butterfly rotations they can correspond to two calls from the same service. Additionally,  $g_{i'i''}$  is the necessary transshipment buffer time between arrival of port call  $i'$  and departure of port call  $i''$  in port  $p$ . The constant  $\bar{t}_i$  specify the length of the stay of port call  $i$  and the continuous decision variable  $\hat{T}_{i'i''} \in \mathbb{R}$  is the transshipment time from port call  $i'$  to port call  $i''$  in a specific port.  $A(i)$  is the set of multiarcs between port call  $i$  and  $i + 1$ , i.e., arcs at different speeds between two consecutive ports on the same rotation.  $(i', i'') \in Q^2(q)$  denotes all the ordered pairs of rotations visiting port  $q$ . For each port  $q$  with a transshipment opportunity and for each  $(i', i'') \in Q^2(q)$ , the set of arcs available for transshipment is given by the set  $A(i', i'')$ . The capacity of the corresponding arc,  $u_{i'i''}$ , is given by the minimum capacity of the rotation corresponding to port call  $i'$  and the rotation corresponding to  $i''$  in a given port. All transshipment arcs have an associated transshipment time,

$t_a$ , and we include binary variables,  $x_a$  for  $a \in A(i', i'')$  that selects whether a transshipment arc is used. The time scheduling part of the model is:

$$T_i = T_{i-1} + \sum_{a \in A(i-1)} t_a x_a \quad i \in I \setminus \cup_{r \in R} \{\sigma_r\} \quad (5.52)$$

$$\hat{T}_{i'i''} = (T_{i''} + \bar{t}_{i''}) - T_{i'} + 7w_{i'i''} \quad q \in Q, (i', i'') \in Q^2(q) \quad (5.53)$$

$$\sum_{a \in A(i'i'')} t_a x_a \geq \hat{T}_{i'i''} \quad q \in Q, (i', i'') \in Q^2(q) \quad (5.54)$$

$$\sum_{a \in A(i'i'')} x_a = 1 \quad q \in Q, (i', i'') \in Q^2(q) \quad (5.55)$$

$$g_{i'i''} \leq \hat{T}_{i'i''} \leq g_{i'i''} + 7 \quad q \in Q, (i', i'') \in Q^2(q) \quad (5.56)$$

$$T_i \in \mathbb{R}_+ \quad i \in I \quad (5.57)$$

$$\hat{T}_{i'i''} \in \mathbb{R}_+ \quad q \in Q, (i', i'') \in Q^2(q) \quad (5.58)$$

$$w_{i'i''} \in \mathbb{Z} \quad q \in Q, (i', i'') \in Q^2(q) \quad (5.59)$$

$$x_a \in \{0, 1\} \quad a \in A(i', i''), q \in Q, (i', i'') \in Q^2(q) \quad (5.60)$$

The relation of the departure time for two consecutive ports is given by constraints (5.52). Notice that we in constraints (5.52) let  $i$  run in the elements of  $I$  except the starting port call of each rotation. The reason is that we need to avoid a cyclic definition of  $T_i$ , which would be infeasible. The transshipment time in port  $p$  from port call  $i'$  to port call  $i''$  is determined by constraints (5.53) and constraints (5.54) and (5.55) makes sure we only select one transshipment arc and that it is feasible. Constraints (5.56) limits the possible transshipment time. If we have a schedule determined by the hour there are going to be 168 transshipment arcs for each feasible  $(i', i'')$ -combination, but we can reduce the number of available transshipment arcs such that we overestimate the transshipment time. There can e.g. be one available arc for each day, i.e., 7 arcs, and for some ports we can have higher accuracy than others by including more arcs.

To illustrate the offset variable consider an instance with hourly accuracy where  $\bar{t}_{i''} = 8$  and  $g_{i'i''} = 10$  then if  $T_{i'} = 24$  and  $T_{i''} = 48$  we get that  $w_{i'i''} = 0$  and  $\hat{T}_{i'i''} = 48 + 8 - 24 = 32$ . If there is a too tight schedule, i.e.,  $T_{i'} = 24$  and  $T_{i''} = 24$ , we get that the commodity will have to wait because of the buffer time and we get that  $w_{i'i''} = 1$  and  $\hat{T}_{i'i''} = 24 + 8 - 24 + 168 = 176$ . To illustrate the influence of the schedule on longer rotations consider first  $T_{i'} = 13 * 24 = 312$  and  $T_{i''} = 24$ , then we get  $w_{i'i''} = 2$  and  $\hat{T}_{i'i''} = 24 + 8 - 312 + 2 * 168 = 56$ .

Conversely if  $T_{i'} = 24$  and  $T_{i''} = 312$ , then we get  $w_{i'i''} = -1$  and  $\hat{T}_{i'i''} = 312 + 8 - 24 - 168 = 128$ .

### 5.8.1.2 Including the Time Scheduling Part in the Benders Decomposition

Similarly to the coupling constraints for the sailing arcs, we can introduce a coupling constraints for the transshipment arcs

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq x_a u_{i'i''} \quad a \in A(i', i''), q \in Q, (i', i'') \in Q^2(q) \quad (5.61)$$

Then including the time scheduling part in the Benders decomposition lead to a slightly modified sub-problem where we are now considering the transshipment arcs as well such that the PBSP is given by:

$$\min \sum_{k \in K} \sum_{p \in P^k} r_p^k y_p^k \quad (5.62)$$

subject to

$$\sum_{p \in P^k} y_p^k \leq d^k \quad k \in K \quad (5.63)$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq \bar{x}_a q_a \quad q \in Q, i \in I(q), a \in A(i) \quad (5.64)$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq \bar{x}_a u_{i'i''} \quad q \in Q, (i', i'') \in Q^2(q), a \in A(i', i'') \quad (5.65)$$

$$y_p^k \in \mathbb{R}_+ \quad k \in K, p \in P^k \quad (5.66)$$

This can still be solved using column generation, but now the column generation sub-problem also contains dual variables corresponding to transshipment arcs. The BMP will be (5.22)-(5.28) + (5.52)-(5.60).

## 5.8.2 Generating Additional Benders Cuts

It is possible to obtain additional Benders cuts in each iteration of the algorithm based on a solution to the multi-commodity flow problem. When the sub-problem has been solved we modify the problem to have an additional constraint restricting the objective to be at least as good as the optimal solution

but maximizing some distance to the solution, e.g.:

$$\max \sum_{k \in K} \sum_{p \in P^k} |\bar{y}_p^k - y_p^k| \quad (5.67)$$

subject to

$$\sum_{p \in P^k} y_p^k \leq d^k \quad k \in K \quad (5.68)$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq q_a x_a \quad (a) \in A \quad (5.69)$$

$$\sum_{k \in K} \sum_{p \in P^k} r_p^k y_p^k \leq \sum_{k \in K} \sum_{p \in P^k} r_p^k \bar{y}_p^k \quad (5.70)$$

$$y_p^k \in \mathbb{R}_+ \quad k \in K, p \in P^k \quad (5.71)$$

where  $\bar{y}_p^k$  is an optimal solution and we use the set of already generated paths to find an alternative solution. However, the objective (5.67) is non-linear so we modify it such that if  $\bar{y}_p^k = 0$  then we take  $(y_p^k - \bar{y}_p^k)$  and if  $\bar{y}_p^k = d^k$  then we take  $(\bar{y}_p^k - y_p^k)$ . If  $0 < \bar{y}_p^k < d^k$  we use  $(\bar{y}_p^k - y_p^k)$  as the objective with probability  $\frac{\bar{y}_p^k}{d^k}$  and  $y_p^k - \bar{y}_p^k$  otherwise. Leaving out the constant term, the objective becomes:

$$\max \sum_{k \in K} \left( \sum_{p \in P^k: \bar{y}_p^k = 0} y_p^k - \left( \sum_{p \in P^k: \bar{y}_p^k = d^k} y_p^k \right) + \sum_{p \in P^k: 0 < \bar{y}_p^k < d^k} X_p y_p^k \right) \quad (5.72)$$

Where  $X_p$  is a random variable of  $\pm 1$  with probability  $\frac{\bar{y}_p^k}{d^k}$ . Using only a reduced set of columns when solving the primal problem will lead to the optimal primal solution, but the corresponding dual solution may not be optimal/feasible as only a subset of constraints are considered. To obtain appropriate dual values, the objective is changed to the original objective, such that constraint (5.70) is removed and the problem is resolved using the solution as a warm start. The new dual solution is checked by solving the pricing problem in multi-commodity flow problem once, and if no reduced cost columns are returned an additional cut is added based on the new dual variables. If a reduced cost column is found, we do not add a cut.

Similarly we could solve the multi-commodity flow problem with the set of columns already obtained with an interior point method to obtain an alternative Benders cut based on a solution centered towards the interior. Again we need to check the multi-commodity flow pricing problem and only add the cut if no reduced cost paths are found.



## Bibliography

- Atamtürk, A. and Rajan, D. (2002). On splittable and unsplittable flow capacitated network design arc-set polyhedra. *Mathematical Programming*, 92(2):315–333.
- Bai, L. and Rubin, P. A. (2009). Combinatorial benders cuts for the minimum tollbooth problem. *Operations research*, 57(6):1510–1522.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252.
- Boccia, M., Sforza, A., Sterle, C., and Vasilyev, I. (2008). A cut and branch approach for the capacitated p-median problem based on fenchel cutting planes. *Journal of Mathematical Modelling and Algorithms*, 7(1):43–58.
- Boyd, E. A. (1994). Fenchel cutting planes for integer programs. *Operations Research*, 42(1):53–64.
- Brouer, B., Álvarez, J., Plum, C., Pisinger, D., and Sigurd, M. (2014a). A base integer programming model and benchmark suite for liner shipping network design. *Transportation Science*, 48(2):281–312.
- Brouer, B., Desaulniers, G., Karsten, C., and Pisinger, D. (2015). A matheuristic for the liner shipping network design problem with transit time restrictions. In Corman, F., Voß, S., and Negenborn, R., editors, *Computational Logistics*, volume 9335 of *Lecture Notes in Computer Science*, pages 195–208. Springer International Publishing.
- Brouer, B. D., Desaulniers, G., and Pisinger, D. (2014b). A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59.
- Cheaitou, A. and Cariou, P. (2012). Liner shipping service optimisation with reefer containers capacity: an application to northern europe–south america trade. *Maritime Policy & Management*, 39(6):589–602.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Christiansen, M., Fagerholt, K., and Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18.

- Cordeau, J.-F., Soumis, F., and Desrosiers, J. (2000). A benders decomposition approach for the locomotive and car assignment problem. *Transportation science*, 34(2):133–149.
- Cordeau, J.-F., Soumis, F., and Desrosiers, J. (2001). Simultaneous assignment of locomotives and cars to passenger trains. *Operations research*, 49(4):531–548.
- Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429–1450.
- Fischetti, M. and Monaci, M. (2014). Proximity search for 0-1 mixed-integer convex programming. *Journal of Heuristics*, 20(6):709–731.
- Fortz, B. and Poss, M. (2009). An improved benders decomposition applied to a multi-layer network design problem. *Operations research letters*, 37(5):359–364.
- Gelareh, S. and Meng, Q. (2010). A novel modeling approach for the fleet deployment problem within a short-term planning horizon. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):76–89.
- Gendron, B. (2011). Decomposition methods for network design. *Procedia-Social and Behavioral Sciences*, 20:31–37.
- Kaparis, K. and Letchford, A. N. (2010). Separation algorithms for 0-1 knapsack polytopes. *Mathematical programming*, 124(1-2):69–91.
- Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138.
- Kontovas, C. and Psaraftis, H. N. (2011). Reduction of emissions along the maritime intermodal container chain: operational models and policies. *Maritime Policy & Management*, 38(4):451–469.
- Magnanti, T. L. and Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations research*, 29(3):464–484.
- Man (2013). Basic principles of ship propulsion. Technical report, Man Diesel & Turbo.
- Meng, Q. and Wang, S. (2011). Optimal operating strategy for a long-haul liner service route. *European Journal of Operational Research*, 215(1):105–114.

- Meng, Q., Wang, S., Andersson, H., and Thun, K. (2014). Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280.
- Notteboom, T. E. and Vernimmen, B. (2009). The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337.
- Psaraftis, H. N. and Kontovas, C. A. (2013). Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C: Emerging Technologies*, 26:331–351.
- Psaraftis, H. N. and Kontovas, C. A. (2015). Slow steaming in maritime transportation: Fundamentals, trade-offs, and decision models. In *Handbook of Ocean Container Transport Logistics*, pages 315–358. Springer.
- Ronen, D. (2011). The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society*, 62(1):211–216.
- Wang, S. and Meng, Q. (2012a). Liner ship fleet deployment with container transshipment operations. *Transportation Research Part E: Logistics and Transportation Review*, 48(2):470–484.
- Wang, S. and Meng, Q. (2012b). Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B: Methodological*, 46(5):615–633.
- Wang, S. and Meng, Q. (2012c). Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):701–714.
- Xia, J., Li, K. X., Ma, H., and Xu, Z. (2015). Joint planning of fleet deployment, speed optimization, and cargo allocation for liner shipping. *Transportation Science*, ( ): .
- Zacharioudakis, P. G., Iordanis, S., Lyridis, D. V., and Psaraftis, H. N. (2011). Liner shipping cycle cost modelling, fleet deployment optimization and what-if analysis. *Maritime Economics & Logistics*, 13(3):278–297.

# Competitive Liner Shipping Network Design

with B.D. Brouer and D. Pisinger<sup>1</sup>

---

## Abstract

We present a solution method for the liner shipping network design problem which is a core strategic planning problem faced by container carriers. We propose the first practical algorithm which explicitly handles time limits for all demands. Individual sailing speeds at each service leg are used to balance sailings speed against operational costs, hence ensuring that the found network is competitive on both transit time and cost. We present a matheuristic for the problem where a MIP is used to select which ports should be inserted or removed on a route. Computational results are presented showing very promising results for realistic global liner shipping networks. Due to a number of algorithmic enhancements, the obtained solutions can be found within the same time frame as used by previous algorithms not handling time constraints. Furthermore we present a sensitivity analysis on fluctuations in bunker price which confirms the applicability of the algorithm.

---

<sup>1</sup>Karsten, C. V., Brouer, B. D., and Pisinger, D. (2015). Competitive Liner Shipping Network Design. Submitted to *Computers and Operations Research*

## 6.1 Introduction

Given a fleet of container vessels and a selection of ports, the classical *Liner Shipping Network Design Problem (LSNDP)* constructs a set of scheduled routes (*services*) with a fixed frequency for container vessels to provide transport for containers worldwide (Brouer et al., 2014a). This paper presents the *Competitive Liner Shipping Network Design Problem (CLSNDP)* extending the classical LSNDP to consider *level of service*, i.e. the transit time provided for a given cargo as well as the transportation cost charged. These two parameters are the main concern for customers, and hence they are crucial parameters for designing competitive networks.

The classical LSNDP is offset in the main objective of the carrier; to maximize profit through the revenues gained from container transport taking into account the fixed cost of deploying vessels and the variable cost related to the operation of the services. The opposing objectives of the customer and the carrier represents an inherent trade-off in the design of a liner shipping network. Minimizing the cost of the network will provide low freight rates, but are likely to result in prolonged transit times as shown by Karsten et al. (2015a). On the other hand, designing a network to minimize transit times is likely to result in a very costly network favoring direct connections at high sailing speeds.

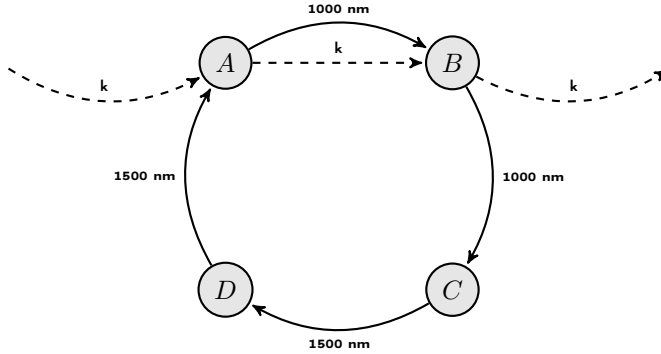
The models for the classical LSNDP differ on two traits. First, the ability to model and charge transshipments between services. Containers are often not transported directly from their port of origin to their port of destination, and hence it is important to be able to handle the time and cost of transshipments. Second, models differ on requiring a fixed frequency of service or providing flexibility in the frequency. A service is cyclic but may be non-simple, that is, ports can be visited more than once. In this model we allow a single port to be visited twice, yielding a so-called *butterfly* route.

The paper by Agarwal and Ergun (2008) imposes a weekly frequency of service and allows for transshipment, but the model cannot cater for the handling cost associated with transshipments. The paper by Álvarez (2009) can cater for transshipment and transshipment costs (except within butterfly services) and allows for flexible frequencies of service. In Reinhardt and Pisinger (2012) each vessel is treated separately allowing flexible frequencies, and the model allows for transshipment costs also on butterfly routes. Brouer et al. (2014a) provides an analysis of the real life requirements and present a reference model for the classical LSNDP. The model is offset in Álvarez (2009) accounting correctly for transshipments on all services and allowing both flexible and fixed frequencies. The above models are all variants of specialized capacitated network design

problems.

Meng et al. (2014); Christiansen and Fagerholt (2011); Christiansen et al. (2013) provide broader reviews of recent research on routing and scheduling problems within liner shipping. In the literature several papers extend the classical LSNP e.g. by incorporating intermodal considerations (Liu et al., 2014) or aiming to narrow the definition of service (Plum et al., 2014). However, it is generally acknowledged that considering level of service is the most important extension to the classical LSNP because it is the decisive factor in designing a competitive network (Álvarez, 2012; Brouer et al., 2014a). Two approaches for considering level of service has been suggested in the literature. The first method is to include inventory cost in a multi-criteria objective function as seen in Álvarez (2012). Inventory cost is primarily a concern to the shipper and the idea of introducing it for the carrier is to ensure that longer transit times will result in lower freight rates. However, the bilinear expression proposed by Álvarez (2012) is not computationally tractable. Another approach is to impose restrictions on the allowed transit times for each container. The idea here is that the carrier needs to provide competitive transit times in a market of several players. Wang and Meng (2014) introduce deadlines on cargo in a non-linear, non-convex mixed-integer programming (MIP) formulation of a LSNP. A drawback of this formulation is that it cannot cater for transshipments of cargo which is the backbone of global liner shipping networks. Recently Brouer et al. (2015) presented a capacitated multi-commodity network design formulation that imposes transit time restrictions while still allowing transshipments between services and Karsten et al. (2015a) showed that time restricted multi-commodity flow problem arising as a sub-problem can be efficiently solved for a large global shipping network. The CLSNP in this paper build upon these contributions.

Introducing transit time restrictions is essential in the LSNP from a customer perspective, but to maintain low fuel (bunker) cost this must be accompanied by modelling the services with variable speed. Traditionally, models of the LSNP operate with a constant speed on services although variable speed on each leg is used in practice. In a network with constant speed the most transit time restricted commodity will force the entire service to speed up, and hence increase the bunker consumption of the service unnecessarily with a resulting increase in both cost and  $CO_2$  emissions. Figure 6.1 illustrates the problem of maintaining constant speed during the design process. The container entering at  $A$  and leaving at  $B$ ,  $k_{AB}$ , has the tightest transit time requirement among the containers currently transported on service  $s$  with a transit time restriction of 3 days, which requires a speed of 14 knots. This results in a deployment of 2 vessels at a speed of nearly 21 knots, because of



**Figure 6.1:** A service illustrated with constant speed and weekly frequency. The nodes are ports and the solid lines correspond to sailing edges. Two deployments are possible to complete the round trip of 5,000 nm (nautical miles) within the speed bounds: Three vessels deployed ( $n_e = 3$ ) results in a constant speed of 12.25 knots, while two vessels deployed ( $n_e = 2$ ) results in a constant speed of 20.83 knots. The most transit time critical commodity,  $k$ , on the service is for the commodity illustrated by the dashed line from  $A$  to  $B$ , where the transit time restriction is 3 days requiring a speed of 14 knots.

only two possible deployments with constant speed and the weekly frequency requirement imposed. If speed can be determined individually on each sailing leg, 3 vessels can be deployed with a speed of 14 knots between  $A$  and  $B$  and a speed of 12 knots on the remaining sailing legs maintaining the weekly frequency but resulting in a significant decrease in the bunker consumption (since the bunker consumption is a cubic function of the speed (Brouer et al., 2014a)). The computational results presented in Brouer et al. (2015) support a higher average speed and low fleet deployment in networks optimized with transit time restrictions and constant speed.

Therefore, the CLSNDP is extending the reference model for LSNDP Brouer et al. (2014a) to consider transit time restrictions coupled with variable speed on each sailing leg in order to properly address the trade-off between providing competitive transit times, while reducing cost as well as  $CO_2$  emissions. In this paper we propose the first algorithm to solve CLSNDP by an adaptation of the matheuristic of Brouer et al. (2014b) that considers transit times and optimize speed on each sailing leg. The underlying basis for the model is a capacitated multi-commodity network design formulation where we can accurately model transshipment operations, cost structures, and restrictions on container transit time of individual containers. The formulation adheres to the objective and constraints of Brouer et al. (2014a) with a fixed weekly frequency. As we are

not solving the mathematical formulation using an exact algorithm we have chosen to place the mathematical model in Appendix 6.6.

Speed optimization in maritime transportation has received quite a lot of interest in the literature across economics and operations research over the past decade. Psaraftis and Kontovas (2013) survey models and taxonomy on speed optimization and in Psaraftis (2015) “slow steaming” as a phenomenon is discussed. Notteboom and Vernimmen (2009) and Ronen (2011) provide insights on speed optimization in liner shipping and show the importance of optimizing speed in liner shipping networks by studying a single service. There are numerous examples of speed optimization within liner shipping e.g. the non-linear MIP formulation presented in Wang and Meng (2012c), or speed optimization coupled with fleet deployment e.g. (Gelareh and Meng, 2010; Meng and Wang, 2011; Zacharioudakis et al., 2011). A number of contributions are concerned with the coupling between transit time and speed in optimizing the network (Cheaitou and Cariou, 2012; Wang and Meng, 2012a,b). Reinhardt et al. (2015) present a MIP model for adjusting the port berth times such that the fuel consumption is minimized while retaining the customer transit times. A penalty is assigned to each change of berth time in order to limit the number of changes. Karsten et al. (2015b) use Benders decomposition to simultaneously optimize sailing speed and container routing. All containers have an associated limit on the transit times that needs to be met.

Deciding an optimal speed configuration in a liner shipping network requires consideration of the network in its entirety as transit times of commodities may be decided by several interoperating services. Likewise commodity paths are likely to change with the speed optimization if cargo routings are flexible. However, computational results from the above mentioned papers indicate that this is not computationally tractable for revaluation in a large-scale heuristic search. The matheuristic for the CLSNDP proposed in this paper is considering speed as one of the dimensions in the solution space and therefore a fast method for optimizing speed is needed. In tramp shipping speed optimization of an isolated route in the network is optimal. Variable speed for a single ship route in tramp shipping has been explored in Fagerholt et al. (2009); I. Norstad and Laporte (2011); Hvattum et al. (2013), where the introduction of speed optimization allowing variable speed on a sail route results in significant fuel savings. In Fagerholt et al. (2009) a MIP with a non-linear objective function depicting the vessels fuel consumption as a function of speed is presented. The speed optimization problem can be transformed into a directed acyclic graph if speeds are discretized and the resulting speed profile is simply a shortest path, which can be efficiently calculated for a directed acyclic graph. The approach by Fagerholt et al. (2009) cannot be adopted directly, since a liner shipping



service will be carrying multiple commodities and hence the time windows are defined per pickup node. Transforming the problem into a graph would result in node specific time windows accounting for times between every OD pair assigned to the service, which would require a resource constrained shortest path with a specific resource for every port in the service. This is unlikely to be efficiently solved. However, we can adapt the non-linear MIP formulation of Fagerholt et al. (2009) to optimize speed on a single service given constraints on the slack time of each commodity currently transported on the service. As a novelty we also consider opportunity cargo not currently transported, as speed optimization may lead to new attractive transport opportunities. The non-linear bunker consumption function is approximated by a piecewise linear function of the time to sail a given leg and the speed optimization MIP can be efficiently solved using a standard MIP solver making it suitable to incorporate into a heuristic. Our computational results show that it is tractable to incorporate level of service in the network design process by considering container transit time restrictions *and* variable speed in a heuristic context, and we are able to design profitable networks for scenarios resembling global liner shipping networks.

The rest of the article is organized as follows. Section 6.2 discusses the extensions from the LSNDP to the CLSNDP. Section 6.3 gives an overview of our solution method and describes the level of service implications in detail. Section 6.4 presents computational results on realistic instances from the benchmark suite *Liner-lib* before we conclude and discuss future work in Section 6.5.

## 6.2 Problem Description

Given a fleet of container vessels and a selection of ports, the CLSNDP constructs a set of services to provide transport for containers worldwide. It extends the classical LSNDP to consider level of service as this is the main concern for the shipper. The CLSNDP we present here is based on the reference model for the LSNDP presented in Brouer et al. (2014a) which has been extended in Brouer et al. (2015) to consider transit time restrictions for all commodities, see Appendix 6.6 for a full description of the model. The primary change in order to accommodate transit time restrictions into the model of Brouer et al. (2014a) is to decompose the multi commodity flow problem into a path flow formulation. In the path flow formulation only paths respecting the maximal transit time for a given commodity are feasible. This extension of the LSNDP with transit time restrictions is a non-compact formulation with integer service variables defining a port call sequence, a vessel type, number of ships and a

constant speed, and real path variables for routing the commodities. As transit times are closely linked to speed, the constant speed needed to accommodate transit time restrictions will generally be determined by the commodity with the most restrictive transit time. However, it is unnecessary to maintain a high speed throughout the service if this commodity is only carried on part of the service. Therefore we use service variables that include variable speed by allowing each sailing to take on any speed within the feasible speed interval, while maintaining a weekly frequency of service. The overall objective of CLSNBP is to maximize profit, however, the extensions potentially result in fuel savings and/or a larger cargo uptake in the network along with ensuring a competitive level of service in the network.

The next section provides a broad overview of the algorithm and its components. The overview includes the extensions necessary to enable consideration of level of service, namely transit time restrictions for each individual commodity and optimizing speed on each sailing in the network. Following the overview the extensions will be described in further detail.

## 6.3 Algorithm

The proposed matheuristic is based on the algorithm from Brouer et al. (2014b). Since the evaluation of the objective function makes it necessary to flow all containers through the network, only a limited number of iterations can be evaluated throughout the search, and therefore it is important to use a large neighborhood search, combined with a shrewd way of choosing the direction of the search.

Algorithm 1 presents high level pseudocode for the overall matheuristic. Initially a solution is constructed by dividing the available fleet onto services. Subsequently the services are populated with port calls following a greedy parallel insertion procedure according to the distance and the trade volume between ports in the service in line 1. The subsequent search for improved solutions is guided by a simulated annealing scheme in the while loop of lines 5-25. The primary component of the matheuristic is a neighbourhood for inserting and removing port calls on a single service which is formulated as an integer program in line 8. The integer program is described in detail in Section 6.3.1. In order to optimize speed in the network a heuristic method based on a non-linear MIP is applied. The heuristic optimizes the speed of all legs on a single service given the time limits of cargo currently transported on this service and the time limit of opportunity demands, that are currently rejected due to transit time

restrictions. This MIP is called in line 10 after resolving the multicommodity flow problem in line 9 given the changes to service  $s$ . As changes are only made to a single service, the column generation algorithm used is warm started using the technique described in Brouer et al. (2014b). The simulated annealing scheme decides whether the new solution is accepted in line 12. The reinsertion heuristic in line 18 introduces butterfly ports on promising candidate services. The perturbation heuristic in line 23 diversifies the service composition. The two latter heuristics are unchanged to the versions in Brouer et al. (2014b).

---

**Algorithm 1** High Level algorithm for CLSNDP

---

**Require:** An instance of the CLSNDP

```

1: Construct an initial solution  $x$  using a greedy algorithm
2: Set the best known solution  $x^* = x$ 
3: Set the iteration counter  $iter = 0$ 
4: Set the initial temperature  $temp = temp_0$ 
5: while  $temp > 0.01$  AND  $time < MAXtime$  do
6:   for each service  $s \in x$  do
7:      $x' \leftarrow x \setminus s$ 
8:      $s' \leftarrow IP(s)$ : improve solution by insertion/removal of port calls on
       service  $s$ 
9:     Resolve cargo flow
10:    Optimize speed of each sailing on  $s'$ 
11:     $x' \leftarrow x' \cup s'$ 
12:    if accept solution according to cooling scheme then
13:      Set  $x \leftarrow x'$ 
14:      Possibly update best known solution:  $x^* \leftarrow x$ 
15:     $iter \leftarrow iter + 1$ 
16:     $temp \leftarrow temp \cdot 0.98$ 
17:    if  $iter \bmod 4 = 0$  then
18:      Apply reinsertion heuristic to obtain new solution  $x'$  with promising
        butterfly routes
19:      if Solution improves then
20:        Set  $x \leftarrow x'$ 
21:        Possibly update best known solution:  $x^* \leftarrow x$ 
22:    if  $iter \bmod 10 = 0$  then
23:      Apply perturbation to obtain a solution  $x'$  with a different service
        composition
24:      Set  $x \leftarrow x'$ 
25:      Possibly update best known solution:  $x^* \leftarrow x$ 
return  $(x^*)$ 

```

---

### 6.3.1 Level of Service Considerations

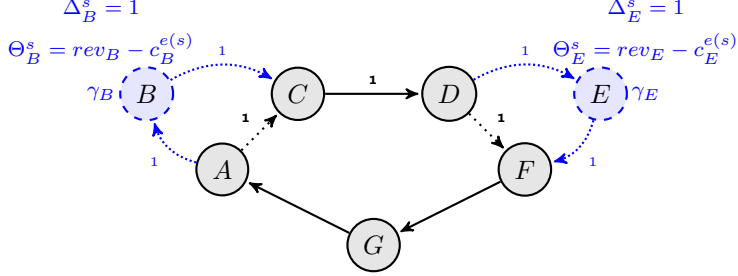
The integer program described in line 8 of Algorithm 1 is a move operator in a large neighborhood search based on altering a single service at a time. The objective of the integer program is an estimation function for changes in the flow of the network and the duration of the service due to insertions and removals of port calls. The solution of the integer program provides a set of moves in the composition of port calls and fleet deployment. Flow changes and the resulting change in the revenue for relevant commodities to the insertion/removal of a port call are estimated by solving a series of resource constrained shortest path problems considering feasibility of transit time restrictions as well as the cost of transport including transshipments.

Given a total estimated change in revenue of  $rev_i$  and port call cost of  $c_i^{e(s)}$  Figure 6.2a illustrates estimation functions for the change in revenue ( $\Theta_i^s$ ) and duration increase ( $\Delta_i^s$ ) for inserting port  $i$  into service  $s$  controlled by the binary variable  $\gamma_i$ . The duration controls the number of vessels needed to maintain a weekly frequency. Figure 6.2b illustrates the estimation functions for the change in revenue ( $\Upsilon_i^s$ ) and decrease in duration ( $\Gamma_i^s$ ) for removing port  $i$  from service  $s$  controlled by the binary variable  $\lambda_i$ .

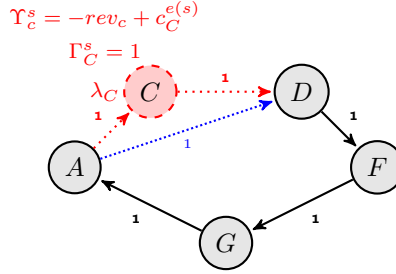
For considering the transit time in the IP, it is necessary to estimate how insertions and removals of port calls will affect the duration of the existing flow on the service. If an insertion is estimated to result in exceeding the transit time restriction of existing flow, and there is no possibility of rerouting the flow on a different path respecting the transit time limits, a loss of revenue can be expected. The loss is estimated to correspond to the full revenue obtained from the demand quantity. Figure 6.3 illustrates a case of a path variable in the current basis of the MCF model, which becomes infeasible due to transit time restrictions when inserting port  $B$  on its path.

In order to account for the transit time restrictions of the current flow, additional constraints are added to the IP and a penalty,  $\zeta_x$  corresponding to losing the cargo, is added to the objective if the transit time slack for an existing path variable becomes negative. This is handled through the variable  $\alpha_x$ , where  $x$  refers to a path variable with positive flow in the current solution and  $s_x$  refers to the current slack time according to the transit time restrictions of the variable. Variable speed is considered in the estimation function for the flow as well as for the estimation of the service duration. The speed on the sailings to and from the port evaluated for insertion is estimated to be equal to the speed sailed between the two ports previously connected and is denoted by the constant  $K_{\gamma_i}$ . Upon evaluating a removal of a port the actual speed of the

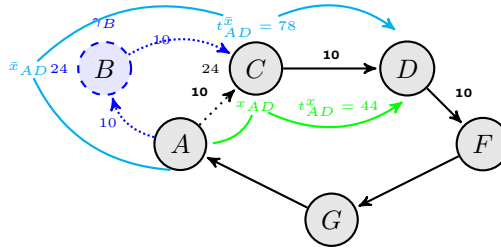
(a) Blue nodes are evaluated for insertion corresponding to variables  $\gamma_i$  for the set of ports in the neighborhood  $N^s$  of service  $s$ .



(b) Red nodes are evaluated for removal corresponding to variables  $\lambda_i$  for the set of current port calls  $F^s$  on service  $s$ .

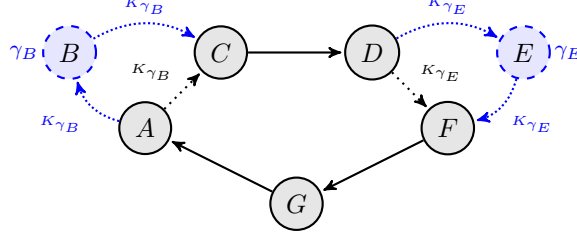


**Figure 6.2:** Illustration of the estimation functions for insertion and removal of port calls.

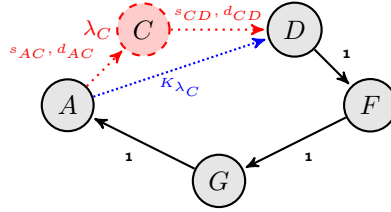


**Figure 6.3:** Insertions/removals affect transit time of the flow. Commodity  $k_{AD}$  has a maximum transit time of 48 hours and the insertion of  $\gamma_B$  will make path variable  $x_{AD}$  infeasible.

(a) Blue nodes are evaluated for insertion corresponding to variables  $\gamma_i$  for the set of ports in the neighborhood  $N^s$  of service  $s$ . Speeds of sailings to and from the insertion correspond to the speed of the existing link.



(b) Red nodes are evaluated for removal corresponding to variables  $\lambda_i$  for the set of current port calls  $F^s$  on service  $s$ . A weighted average speed is used  $K_{\lambda_C} = \frac{d_{AC}}{d_{AC}+d_{CD}} \cdot s_{AC} + \frac{d_{CD}}{d_{AC}+d_{CD}} \cdot s_{DC}$ .



**Figure 6.4:** Illustration of the speeds used by estimation functions for insertion and removal of port calls.

sailing in question is used to reduce the duration of the service. The constant  $K_{\lambda_i}$  expresses the weighted average speed of the current speeds for the sailings entering and leaving the port estimated for removal. The speeds used for the estimation functions are illustrated in Figure 6.4.

For ease of reading, Table 6.1 gives an overview of additional sets, constants, and variables used in the IP.

Sets	
$F^s$	Set of port calls in $s$
$N^s$	Set of neighbors (potential port call insertions) of $s$
$X^s$	Set of path variables on service $s$ in current solution with positive flow
$N^x \subseteq N^s$	Subset of neighbors with insertion on current path of variable $x \in X^s$
$F^x \subseteq F^s$	Subset of port calls on current path of variable $x \in X^s$
$L_i$	Lock set for port call insertion $i \in N^s$ or port call removal $i \in F^s$
Constants	
$Y^s$	Distance of the route associated with $s$
$B_i$	Berthing time for port call $i \in F^s$
$V^s$	Estimated weighted average speed over all sailings on the service $s$
$V_{\gamma_i}$	Speed between insertion points on the service $s$
$V_{\lambda_i}$	Speed on sailing removed from the service $s$
$C_e$	Cost of an additional vessels of class $e(s)$
$n_e$	Number of deployed vessels of class $e(s)$ to $s$ in the current solution
$M_e$	Number of undeployed vessels of class $e$ in the current solution
$I^s$	Maximum number of insertions allowed in $s$
$R^s$	Maximum number of removals allowed in $s$
$\Delta_i^s$	Estimated distance increase if port call $i \in N^s$ is inserted in $s$
$\Gamma_i^s$	Estimated distance decrease if port call $i \in F^s$ is removed from $s$
$\Theta_i$	Estimated profit increase of inserting port call $i \in N^s$ in $s$
$\Upsilon_i$	Estimated profit increase of removing port call $i \in F^s$ from $s$
$\zeta_x$	Estimated penalty for cargo lost due to transit time
$s_x$	Slack time of path variable $x$
Variables	
$\lambda_i$	Binary, 1 if port call $i \in F^s$ is removed from $s$ , 0 otherwise
$\gamma_i$	Binary, 1 if port call $i \in N^s$ is inserted in $s$ , 0 otherwise
$\omega^s$	Integer, number of vessels added (removed if negative) to $s$
$\alpha_x$	Binary, 1 if transit time of path variable $x \in X^s$ is violated, 0 o.w.

Table 6.1: Overview of sets, constants, and variables used in the IP.

The objective of the move operator is to maximize the estimated profit increase obtained from removing and inserting port calls, accounting for the estimated change of revenue, transshipment cost, port call cost, and fleet cost:

$$\max \sum_{i \in N^s} \Theta_i \gamma_i + \sum_{i \in F^s} \Upsilon_i \lambda_i - C_e \omega^s - \zeta_x \alpha_x \quad (6.1)$$

First, we need to estimate the number of vessels  $\omega^s$  needed on the service  $s$  (assuming a weekly frequency) after insertions/removals while accounting for the change in the service time given the current weighted average speed on the service  $V^s$ :

$$\frac{Y^s}{V^s} + \sum_{i \in F^s} B_i + \sum_{i \in N^s} \left( \frac{\Delta_i^s}{V_{\gamma_i}} + B_i \right) \gamma_i - \sum_{i \in F^s} \left( \frac{\Gamma_i^s}{V_{\lambda_i}} + B_i \right) \lambda_i \leq 24 \cdot 7 \cdot (n_e + \omega^s) \quad (6.2)$$

Next, we must ensure that the solution does not exceed the available fleet of vessels. Note that  $\omega^s$  does not need to be bounded from below by  $-n_e$  because it is not allowed to remove all port calls:

$$\omega^s \leq M_e \quad (6.3)$$

Then, a limit on the number of port call insertions and removals is enforced in order to minimize the error in the computed estimates:

$$\sum_{i \in N^s} \gamma_i \leq I^s \quad (6.4)$$

$$\sum_{i \in F^s} \lambda_i \leq R^s \quad (6.5)$$

Furthermore, the flow estimates are based on cargo flowing to and from a set of related port calls on the service. The affected ports are placed in a lock set,  $L_i$ , for insertions and removals respectively, i.e. ports in a lock set cannot be removed to avoid large deviations in the flow estimates:

$$\sum_{j \in L_i} \lambda_j \leq |L_i|(1 - \gamma_i) \quad i \in N^s \quad (6.6)$$

$$\sum_{j \in L_i} \lambda_j \leq |L_i|(1 - \lambda_i) \quad i \in F^s \quad (6.7)$$

Finally, we need to activate the estimated penalty for lost cargo due to an estimated violation of the transit time for the commodity on this particular path:

$$\sum_{i \in N^x} \left( \frac{\Delta_i^s}{V^s} + B_i \right) \gamma_i - \sum_{i \in F^x} \left( \frac{\Gamma_i^s}{V^s} + B_i \right) \lambda_i - UB \alpha_x \leq s_x \quad x \in X^s \quad (6.8)$$



The domains of the variables are:

$$\lambda_i \in \{0, 1\}, i \in F^s \quad \gamma_i \in \{0, 1\}, i \in N^s \quad \alpha_x \in \{0, 1\}, x \in X^s \quad \omega^s \in \mathbb{Z}, s \in S$$

As opposed to the move operator proposed in Brouer et al. (2014b) the change in revenue may be related to not transporting cargo for which the path duration is estimated to exceed the transit time of the commodity.

### 6.3.2 Variable Speed on Service Legs

To include variable speed in the matheuristic (Algorithm 1 line 10) we formulate the speed optimization problem as a mixed integer program with a non-linear objective function that can easily be solved for each service  $s \in S$  during the iterative search.  $m$  is the number of port calls in the round trip of  $s$  and  $m + 1$  is the first port of call. The function  $g(t_{j,j+1}, d_{j,j+1})$  represents the bunker consumption from port  $j$  to  $j + 1$  expressed as a function of sailing time  $t_{j,j+1}$  and distance  $d_{j,j+1}$ , which indirectly models the speed  $v_{j,j+1}$ . For each service we wish to determine the sailing speed of each sailing leg which we do by finding the optimal sailing time  $t_{j,j+1}$  between ports  $j$  and  $j + 1$ . We arrive in port  $j$  at time  $t_j$  and the sailing time must be determined such that the weekly frequency of a service is maintained. If the sailing speed is changed significantly it is possible to add or remove an additional vessel to the service provided that additional vessels are available. As a novelty we also consider commodities that are not currently transported but could be transported on service  $s$  if a sufficient speed increase is profitable. To find the set of candidate commodities for a service we solve an unconstrained shortest path problem on the residual capacity graph of the current network for all commodities that are not currently transported. We add the ones that have a profitable path through service  $s$  to the set but where transit time is then violated to  $K_{p,s}$  and calculate the potential profit based on the residual capacity (which may be less than the demand of a cargo), the cost of the path and the service penalty (which we potentially can avoid). Additionally we keep track of the time decrease needed (corresponding to a speed up) to make the path feasible. The constants, sets and variables used in the model for a specific service  $s \in S$  are summarized in Table 6.2.

Using this notation, the objective for each service is to minimize the objective function accounting for the bunker cost, the expected loss of revenue due to transit times not met and the deployment cost of additional vessels less the profit from demand that become available for transport by adjusting the speed.

**Sets**


---

$\tilde{K}_s$	Set of commodities currently transported on $s$ where $t_{o_k} < t_{d_k}$
$\tilde{\tilde{K}}_s$	Set of commodities currently transported on $s$ where $t_{o_k} > t_{d_k}$
$K_{p,s}$	Set of commodities potentially transported on $s$ where $t_{o_k} < t_{d_k}$
$\tilde{\tilde{K}}_{p,s}$	Set of commodities potentially transported on $s$ where $t_{o_k} > t_{d_k}$

---

**Constants**


---

$T_{min}$	Time to complete service $s$ at minimum speed
$t_k^s$	Time commodity $k$ currently uses on service $s$ and the possible slack time between the time of the current path and the overall transit time limit of $k$
$z_k$	Net revenue that will be lost if not transporting the demand $k \in K_s \cup \tilde{\tilde{K}}_s$
$r_k$	Net revenue obtained by transporting all of demand $k \in K_{p,s} \cup \tilde{\tilde{K}}_{p,s}$
$t_{s,k}^{cur}$	Time commodity $k \in K_{p,s} \cup \tilde{\tilde{K}}_{p,s}$ currently would spend on service $s$
$t_{s,k}^{lack}$	Time currently lacking for commodity $k \in K_{p,s} \cup \tilde{\tilde{K}}_{p,s}$

---

**Variables**


---

$t_j$	Continuous, arrival time at port $j$
$t_{j,j+1}$	Continuous, sailing time between ports $j$ and $j+1$
$\delta_e$	Integer, change in the number of vessels of class $e(s)$ deployed to service $s$
$\rho_k$	Binary, 1 if commodity $k$ will be lost due to transit time violation
$\eta_k$	Binary, 1 if commodity $k$ will be available if transit time is reduced

---

**Table 6.2:** Overview of sets, constants, and variables used in the Speed MIP.

The objective can be written as:

$$\min \sum_{j=1}^m c_{Bg}(t_{j,j+1}, d_{j,j+1}^{e(s)}) + \sum_{K_s \cup \tilde{\tilde{K}}_s} z_k \rho_k + C_e \delta_e - \sum_{K_{p,s} \cup \tilde{\tilde{K}}_{p,s}} r_k \eta_k \quad (6.9)$$

A number of constraints need to be satisfied: First, we need to set the time for each port on a route and the sailing time between ports for calculating the bunker consumption:

$$t_{j+1} - t_j - t_{j,j+1} \geq B_j \quad j = 1, \dots, m \quad (6.10)$$

Next, we decide the number of vessels needed to maintain a weekly frequency on the service including berthing time for each port call:

$$t_{m+1} - 168 \cdot \delta_V = 168 \cdot n_e - \sum_{j=1}^m B_j \quad (6.11)$$

The service time is set by the constraint:

$$\sum_{j=1}^m t_{j,j+1} = t_{m+1} \quad (6.12)$$

Moreover, we invoke a loss of revenue if the transit times of commodities on board the service  $s$  are not met. A separate constraint is necessary for commodities where  $t_{o_k} < t_{d_k}$  to account for the total round trip time:

$$t_{d_k} - t_{o_k} - \rho_k T_{min} \leq t_k \quad k \in K_s \quad (6.13)$$

$$t_{d_k} - t_{o_k} - \rho_k T_{min} + t_{m+1} \leq t_k \quad k \in \tilde{K}_s \quad (6.14)$$

Similar constraints allow a service to pick-up additional cargo if speed is increased sufficiently to make paths for cargo that was previously rejected due to transit time limits:

$$t_{d_k} - t_{o_k} - (1 - \eta_k) T_{min} \leq t_{s,k}^{cur} - t_{s,k}^{lack} \quad k \in K_{p,s} \quad (6.15)$$

$$t_{d_k} - t_{o_k} - (1 - \eta_k) T_{min} + t_{m+1} \leq t_{s,k}^{cur} - t_{s,k}^{lack} \quad k \in \tilde{K}_{p,s} \quad (6.16)$$

Finally, we need to enforce speed bounds of the vessel class used by service  $s$ :

$$t_{j,j+1} \geq \frac{d_{j,j+1}}{v_{max}} \quad j = 1, \dots, m \quad (6.17)$$

$$t_{j,j+1} \leq \frac{d_{j,j+1}}{v_{min}} \quad j = 1, \dots, m \quad (6.18)$$

The variable  $\delta_e$  is bounded from above by the number of available vessels if the service slows down overall by adding an additional vessel to the service. The bounds on  $\delta_e$  are tightened in order to give a good solution close to the current deployment such that  $-1 \leq \delta_V \leq \min\{1, M_e\}$ , i.e. it is only possible to add or remove at most one vessel. The variable domains are:

$$\delta_e \in \{-1, 0, \min\{1, M_e\}\} \quad (6.19)$$

$$t_j, t_{j,j+1} \in \mathbb{R}^+ \quad j = 1, \dots, m \quad (6.20)$$

$$\rho_k \in \{0, 1\} \quad k \in K_s \cup \tilde{K}_s \quad (6.21)$$

$$\eta_k \in \{0, 1\} \quad k \in K_{p,s} \cup \tilde{K}_{p,s} \quad (6.22)$$

The objective function can be linearized by modeling the bunker consumption as a piecewise linear function for each  $t_{j,j+1}$  and the model (6.9)-(6.22) can be solved efficiently by a standard mixed integer programming solver. We use 100 pieces to accurately model the bunker consumption function (the solution times for the speed optimization problem are generally less than 0.1 seconds in the instances we have solved in Section 6.4 and the number of pieces used to approximate the objective only has limited impact on this.)

As described earlier, when a service in the network is changed we re-solve the cargo flowing sub-problem using a warm starting procedure where previously generated columns are used leading to a very effective solution of the flow

problem. It should be noted that solving the speed optimization for each service separately leads to a sub-optimal configuration of the network as a significant portion of the demands uses more than one service and hence the transit time for each demand is determined by more than one service, but as we solve the problem many times for each service as part of the search procedure large differences can be reduced.

## 6.4 Computational Results

The matheuristic was tested on data from the benchmark suite *Liner-lib* described in Brouer et al. (2014a), <http://www.linerlib.org>. Table 6.3 gives an overview of the instances. The transit time restrictions have been updated according to the most recent published liner shipping transit times for a small number of the origin-destination pairs as described in Brouer et al. (2015).

Category	Instance and description	$ \mathbf{P} $	$ \mathbf{K} $	$ \mathbf{E} $	min $v$	max $v$
Single-hub instances	<b>Baltic</b> Baltic Sea, Bremerhaven as hub	12	22	2	5	7
	<b>WAF</b> West Africa, Algeciras as hub	19	38	2	33	51
Multi-hub instance	<b>Mediterranean</b> Mediterranean Sea, Algeciras, Tangier, and Gioia Tauro as hubs	39	369	3	15	25
Trade-lane instances	<b>Pacific</b> Asia - US West Coast	45	722	4	81	119
	<b>AsiaEurope</b> Europe, Middle East, and Far East regions	111	4,000	6	140	212
World instances	<b>Small</b> 47 main ports worldwide identified by Maersk Line	47	1,764	6	209	317

**Table 6.3:** The instances of the benchmark suite with indication of the number of ports ( $|\mathbf{P}|$ ), the number of origin-destination pairs ( $|\mathbf{K}|$ ), the number of vessel classes ( $|\mathbf{E}|$ ), the minimum (min  $v$ ) and maximum number of vessels (max  $v$ ).

The matheuristic has been coded in C++ and run on a linux system with an *Intel(R) Xeon(R) X5550* CPU at *2.67GHz* and *24 GB* RAM. The algorithm is set to terminate after the time limits imposed in Brouer et al. (2014a) if the stopping criterion of the embedded simulated annealing procedure is not fulfilled at the time limit.

We fix the berthing time,  $B_p$  to 24 hours for all ports as in Brouer et al. (2014a)

and the transshipment time,  $t_a$  is fixed to 48 hours for every connection as the concrete time schedule is not known at this stage. The bunker price is set to \$ 600 per ton as in Brouer et al. (2014a). Prices for bunker have nearly halved in the past five years, and to this end Section 6.4.2 is a case study of key performance indicators for networks constructed with bunker prices ranging from \$ 150 to \$ 700 per ton.

### 6.4.1 Computational Results for *Liner-lib*

Table 6.4 shows the performance of the algorithm on the six instances described in Table 6.3. For each instance the performance of the algorithm is shown when the networks are designed with constant and variable speed. We evaluate the average performance of ten networks in the two settings and also report the best found network. In both the constant speed and variable speed setting the algorithm can find profitable solutions (negative objective values) for Baltic, WAF, WorldSmall, and AsiaEurope. The Pacific instance yields unprofitable solutions though both fleet deployment and transported cargo volume is high. For all instances except the single-hub instances the networks generated with variable speed are consistently better than the constant speed network with an improvement of up to 10 % for the average values and up to a more than 60 % better objective value for the best Pacific network. On average around 85 % to 95 % of the available cargo volume is transported except in the Mediterranean instance. Generally the constant speed instances transport slightly more of the cargo volume than the networks operating at variable speed and the fleet deployment is significantly higher for networks operating at variable speed suggesting overall slower sailing speed. This is also evident from Table 6.5 where the weighted average speed for each vessel class is shown for networks with constant and variable speed. Most of the vessel classes sail significantly slower for the larger networks and variable speed networks generally operate around or below design speed whereas the networks with constant speed operate at or in some cases much above design speed.

Table 6.6 gives statistics on the rejected cargo in the networks with variable speed. The reasons for cargo to be rejected is that there are no cargo paths that meet transit time restrictions, that there is no residual capacity or that the origin-destination pair is not connected in the graph. For Baltic, WAF, and Mediterranean cargo is primarily rejected because the corresponding origin-destination pairs are not connected. This indicates that there is a set of ports that the algorithm assesses to be unprofitable to call. For Pacific, WorldSmall, and AsiaEurope cargo is mainly not transported because of transit times that cannot be met but also to a large degree because of lacking capacity. For these

Instance	Obj. Val.	Deployment	Transp. Vol.	CPU Time	
	$\mathbf{Z}(\mathbf{7})$	$\mathbf{D}(\mathbf{v})$ (%)	$\mathbf{D}( \mathbf{E} )$ (%)	$\mathbf{T}(\mathbf{v})$ (%)	(S)
Baltic					
Best (constant speed)	$-1.41 \cdot 10^4$	100	100	87.4	101
Average (constant speed)	$7.45 \cdot 10^4$	100	100	86.7	108
Best (variable speed)	$-0.46 \cdot 10^4$	100	100	87.9	144
Average (variable speed)	$17.4 \cdot 10^4$	100	100	85.1	115
WAF					
Best (constant speed)	$-5.59 \cdot 10^6$	83.3	85.7	97.0	255
Average (constant speed)	$-4.87 \cdot 10^6$	83.3	85.2	94.3	354
Best (variable speed)	$-5.48 \cdot 10^6$	97.2	97.6	97.6	362
Average (variable speed)	$-4.89 \cdot 10^6$	86.2	87.6	91.7	396
Mediterranean					
Best (constant speed)	$2.42 \cdot 10^6$	91.9	95.0	86.9	710
Average (constant speed)	$2.70 \cdot 10^6$	90.5	94.0	78.9	737
Best (variable speed)	$2.19 \cdot 10^6$	91.9	95.0	83.8	1,200
Average (variable speed)	$2.65 \cdot 10^6$	92.5	95.0	79.8	1,200
Pacific					
Best (constant speed)	$3.05 \cdot 10^6$	95.0	91.0	93.3	3,600
Average (constant speed)	$3.65 \cdot 10^6$	94.0	91.9	94.0	3,600
Best (variable speed)	$1.13 \cdot 10^6$	98.2	97.0	90.3	3,600
Average (variable speed)	$3.44 \cdot 10^6$	97.0	96.0	89.5	3,600
WorldSmall					
Best (constant speed)	$-3.54 \cdot 10^7$	82.0	85.2	91.1	10,800
Average (constant speed)	$-3.15 \cdot 10^7$	82.3	85.4	90.9	10,800
Best (variable speed)	$-4.05 \cdot 10^7$	90.5	96.6	89.1	10,800
Average (variable speed)	$-3.48 \cdot 10^7$	90.3	95.8	88.0	10,800
AsiaEurope					
Best (constant speed)	$-1.67 \cdot 10^7$	84.6	90.9	88.8	14,400
Average (constant speed)	$-1.45 \cdot 10^7$	83.9	91.9	88.5	14,400
Best (variable speed)	$-1.88 \cdot 10^7$	94.4	96.0	85.6	14,400
Average (variable speed)	$-1.52 \cdot 10^7$	94.0	96.8	84.9	14,400

**Table 6.4:** Best and average of 10 runs. Results with constant and variable speed. Weekly objective value ( $\mathbf{Z}(\mathbf{7})$ ); percentage of fleet deployed as a percentage of the total volume  $\mathbf{D}(\mathbf{v})$  and as a percentage of the number of ships  $\mathbf{D}(|\mathbf{E}|)$ .  $\mathbf{T}(\mathbf{v})$  is the percentage of total cargo volume transported and  $(\mathbf{S})$  is the execution time in CPU seconds.

Instance	Vessel Class					
	F450	F800	P1200	P2400	PostP	SuperP
Baltic						
Constant Speed	10.8	13.7				
Variable Speed	11.1	13.9				
WAF						
Constant Speed	11.5	13.2				
Variable Speed	10.8	11.7				
Mediterranean						
Constant Speed	11.9	13.7	13.9			
Variable Speed	11.7	13.0	15.5			
Pacific						
Constant Speed	12.0	14.2	15.9	18.2		
Variable Speed	11.2	12.4	14.9	15.6		
WorldSmall						
Constant Speed	12.7	15.5	17.5	19.4	19.4	18.2
Variable Speed	12.0	13.2	16.4	16.4	15.8	15.6
AsiaEurope						
Constant Speed	11.7	13.7	16.5	18.0	19.7	17.6
Variable Speed	11.5	12.8	16.1	14.8	16.6	15.8
Class Characteristics						
Design Speed	12.0	14.0	18.0	16.0	16.5	17.0
Max Speed	14.0	17.0	19.0	22.0	23.0	22.0

**Table 6.5:** Weighted average speed per vessel class over ten runs. The last two rows indicate the design speed and max speed of the corresponding vessel class. F is Feeder, P is Panamax.

only around 25 % is rejected because of no connections. Generally for the cargo that is rejected because of no connection the percentage of rejected demands in terms of number of demands (k) compared to the volume (v) not connected show that there is a lot of low volume cargo here. Further inspection shows that these demands often are from smaller feeder ports where the total available volume is very low which is why they are assessed to be unprofitable by the algorithm.

Instance	Total rejected		Transit time		Capacity		Transit time and capacity		Not connected		
	R	FFE	tt(k)	tt(v)	C(k)	C(v)	ttC(k)	ttC(v)	L(k)	L(v)	
			(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	
Baltic	$\mu$	8	732	1.1	0.2	22.6	77.1	0.0	0.0	76.3	22.7
	$\sigma$	1	164	3.5	0.6	11.5	10.4	0.0	0.0	14.1	10.6
WAF	$\mu$	8	712	7.0	1.2	14.0	26.1	1.7	0.1	77.3	72.6
	$\sigma$	2	314	12.1	2.2	9.7	25.3	5.3	0.3	13.5	24.9
Mediterranean	$\mu$	107	1,527	35.3	50.0	0.2	0.4	4.3	4.0	60.1	45.7
	$\sigma$	8	250	7.2	9.6	0.7	1.0	4.6	3.9	5.9	8.6
Pacific	$\mu$	240	4,657	51.5	34.4	7.8	27.4	13.3	29.9	27.3	8.3
	$\sigma$	23	641	6.7	7.4	3.3	12.6	4.1	11.8	5.9	3.4
WorldSmall	$\mu$	325	15,334	35.8	40.2	19.9	16.7	21.1	23.9	23.2	19.2
	$\sigma$	45	1,872	6.5	8.3	9.4	8.3	11.1	11.7	20.4	43.5
EuropeAsia	$\mu$	1,029	11,597	41.9	44.9	8.4	14.3	21.3	26.4	28.4	14.4
	$\sigma$	97	1,008	7.5	8.3	2.9	3.5	5.8	7.3	8.5	6.3

**Table 6.6:** Statistics on the rejected demand reporting average ( $\mu$ ) and standard deviation ( $\sigma$ ) over ten runs.  $|\mathbf{R}|$  is the number of rejected OD pairs and **FFE** is the corresponding rejected volume;  $\mathbf{tt}(\mathbf{k})$  is the percentage of OD pairs rejected due only to transit time and  $\mathbf{tt}(\mathbf{v})$  is the corresponding percentage of the total volume;  $\mathbf{C}(\mathbf{k})$  is the percentage of OD pairs rejected due only to lack of capacity and  $\mathbf{C}(\mathbf{v})$  is corresponding percentage of the total volume;  $\mathbf{ttC}(\mathbf{k})$  is the percentage of OD pairs rejected due to both transit time and lack of capacity and  $\mathbf{ttC}(\mathbf{v})$  is the corresponding percentage of the total volume;  $\mathbf{L}(\mathbf{k})$  is the percentage of OD pairs not connected and  $\mathbf{L}(\mathbf{v})$  is the corresponding percentage of the total volume.

### 6.4.2 Sensitivity to Bunker Price

The price of bunker is very decisive for the cost of the network and the soaring oil prices of more than 600 \$ per ton seen at the beginning of this decade along with a surplus of capacity in the market gave rise to the “slow-steaming” era. Recently, oil prices have been plummeting to less than 300 \$ per ton, which means that the trade-off between slow steaming by deploying extra vessels and speeding up services is shifting. This section concerns the performance of the algorithm with a varying price of bunker. The test is performed on several WorldSmall instances, where we are using the same initial solutions for different bunker prices. The subsequent improvement heuristic will be highly dependent on the bunker price in evaluating a given move and the best found solutions will potentially differ significantly. We compare solutions for bunker prices in the range from \$ 150 to \$ 700 per ton in terms of vessel deployment, the percentage of cargo transported, and the weighted average speed of the



network.

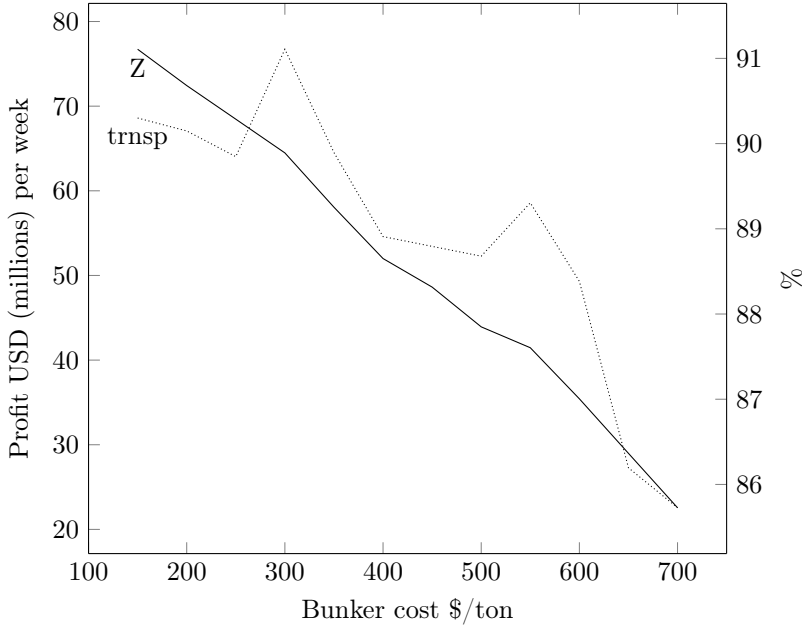
Bunker	Obj. Val.	Deployment		Transp. Vol.
Price	$\mathbf{Z}(\mathbf{7})$	$\mathbf{D}(\mathbf{v})$	$\mathbf{D}( \mathbf{E} )$	$\mathbf{T}(\mathbf{v})$
(\$/ton)	(\$)	(%)	(%)	(%)
150	$7.67 \cdot 10^7$	91.8	95.6	90.3
200	$7.24 \cdot 10^7$	90.2	95.1	90.1
250	$6.85 \cdot 10^7$	91.0	95.3	89.8
300	$6.45 \cdot 10^7$	93.5	96.3	91.1
350	$5.81 \cdot 10^7$	94.4	95.9	89.9
400	$5.20 \cdot 10^7$	91.3	96.3	88.9
450	$4.86 \cdot 10^7$	95.0	97.3	89.3
500	$4.39 \cdot 10^7$	95.0	97.4	88.7
550	$4.15 \cdot 10^7$	94.8	96.9	89.3
600	$3.54 \cdot 10^7$	93.0	96.0	88.4
650	$2.90 \cdot 10^7$	91.5	96.2	86.2
700	$2.26 \cdot 10^7$	93.7	96.7	85.7

**Table 6.7:** Bunker price and the development in the objective value  $\mathbf{Z}(\mathbf{7})$ , deployment percentage of volume  $\mathbf{D}(\mathbf{v})$  and number of vessels  $\mathbf{D}(|\mathbf{E}|)$  and the percentage of cargo transported  $\mathbf{T}(\mathbf{v})$ . Average of five different runs.

Table 6.7 and Figure 6.5 show the correlation between bunker price and the profit margin, which is decreasing with increasing bunker prices. Furthermore, it can be seen that the amount of available cargo transported only decrease a few percent with more then a quadrupling of the bunker price.

Table 6.8 gives statistics on the rejected cargo as a function of bunker price. In Table 6.9 and Figure 6.6 the expected trend of a decreasing speed with an increasing bunker price is clear for all vessel classes except the SuperP class. The weighted average speed confirms this trend. Also, Figure 6.6 shows how the overall deployment is increased when the speed is decreased. The algorithm performs as expected under varying conditions and confirms that even under very different economics conditions we can design profitable networks. The characteristics in terms of deployment and sailing speed of these networks is rather different, but in all cases the algorithm is able to design networks with a high transportation percentage. It should be noted that in these tests only the bunker price is varied while in a real setting the freight rates also depend on the bunker price leading to different network characteristics. However, the sensitivity analysis illustrates how the algorithm also can be used as a managerial tool to conduct “what if” analyses at a strategic level.

The red trend lines in Figure 6.6 show linear fits of the speed ( $f(x) = -0.002x +$



**Figure 6.5:** Development in objective value,  $Z$  (left y-axis), and cargo transported in percentage of total available,  $\text{trnsp}$  (right y-axis), with increasing bunker price. The results are an average of five runs.

16.8), deployment ( $f(x) = 0.002x + 95.2$ ), and amount of transported cargo ( $f(x) = -0.008x + 92.2$ ). These linear approximations confirm the expectation that speed decrease with increased bunker price (0.2 nm/h per 100 \$/ton increase), the amount transported decrease with increased bunker price (0.8 % per 100 \$/ton increase), and deployment increase with increased bunker price (0.2 % per 100 \$/ton increase). This is expected as the bunker consumption is cubic in speed and as the price increase we need more vessels as the network is operating at lower speeds. This also implies that some demands can not meet their transit times even with different service layouts.

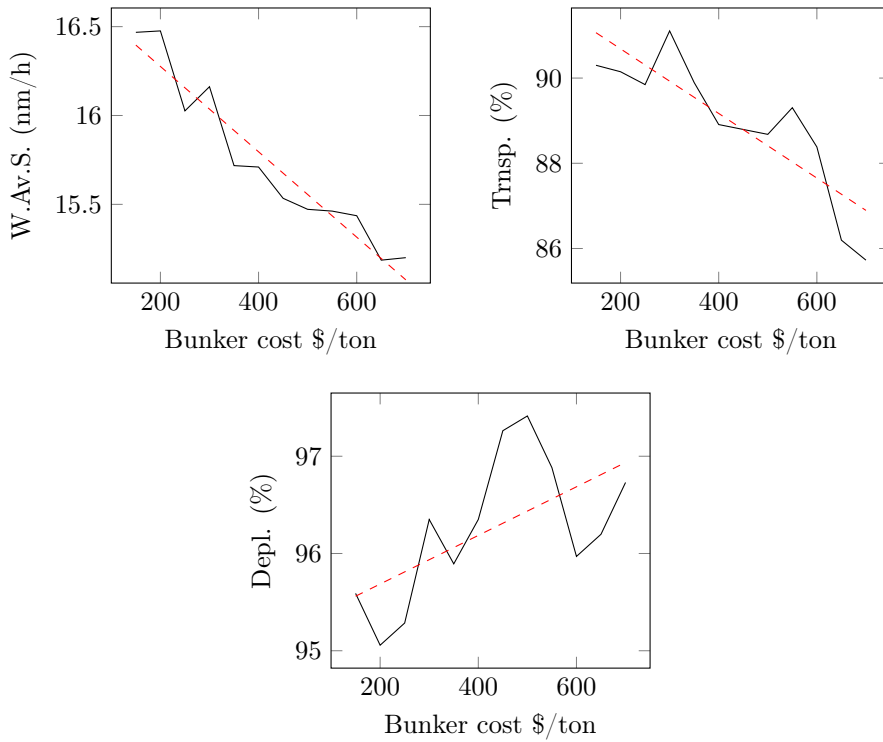
The sensitivity analysis illustrates how the incentives towards slow steaming for liner shipping companies change with varying bunker prices. It will be a more active choice to maintain a greener profile in periods with low oil prices as attaining “an acceptable environmental performance in the transportation supply chain, while at the same time respecting traditional economic performance criteria” (Psaraftis, 2015) is only a win-win solution when oil prices are high.

Bunker price	Total rejected	Transit time		Capacity		Transit time and capacity		Not connected		
(\$/ton)	R	FFE	tt(k) (%)	tt(v) (%)	C(k) (%)	C(v) (%)	ttC(k) (%)	ttC(v) (%)	L(k) (%)	L(v) (%)
150	280	12,443	33.0	37.4	21.9	21.1	16.5	19.3	28.6	22.2
200	264	12,638	41.2	48.3	31.3	24.3	17.7	21.1	9.8	6.4
250	281	13,025	38.6	45.0	22.9	17.9	18.3	22.3	20.3	14.8
300	254	11,408	43.5	49.4	25.4	21.2	21.9	23.3	9.2	6.1
350	277	12,963	47.1	48.1	27.7	21.1	20.7	27.7	4.5	2.9
400	305	14,228	49.4	55.6	15.4	11.6	13.5	17.6	21.8	15.3
450	295	13,776	38.6	41.2	22.5	17.0	21.7	30.0	17.2	11.8
500	303	14,523	50.0	52.4	21.7	18.2	17.3	21.9	10.9	7.3
550	299	13,720	43.5	44.0	24.7	20.9	27.8	31.8	3.9	3.1
600	319	14,902	40.2	42.0	15.5	15.5	21.3	26.6	23.0	15.9
650	374	17,709	53.3	61.0	18.7	13.3	16.3	17.9	11.7	7.9
700	382	18,310	46.0	50.3	18.2	18.3	18.6	20.7	17.3	10.7

**Table 6.8:** Rejected demand given the difference in bunker price.  $|R|$  is the number of rejected OD pairs and **FFE** is the corresponding rejected volume; **tt(k)** is the percentage of OD pairs rejected due only to transit time and **tt(v)** is the corresponding percentage of the total volume; **C(k)** is the percentage of OD pairs rejected due only to lack of capacity and **C(v)** is corresponding percentage of the total volume; **ttC(k)** is the percentage of OD pairs rejected due to both transit time and lack of capacity and **ttC(v)** is the corresponding percentage of the total volume; **L(k)** is the percentage of OD pairs not connected and **L(v)** is the corresponding percentage of the total volume. The results are an average of five runs.

\$/ton	F450	#v	F800	#v	P1200	#v	P2400	#v	PostP	#v	SuperP	#v	Total V	W. Av. S.
150	11.8	24	14.0	29	17.2	66	17.8	74	17.3	53	16.8	7	251	16.5
200	11.9	24	13.5	29	17.1	67	17.8	74	17.5	50	14.0	7	250	16.5
250	11.8	24	13.3	29	16.7	67	17.2	72	16.9	53	13.0	6	251	16.0
300	11.9	24	13.2	28	16.6	65	17.6	74	16.8	55	18.6	7	253	16.2
350	12.2	24	13.3	29	16.4	64	16.6	73	16.6	53	16.2	9	252	15.7
400	11.5	24	13.7	29	16.4	68	16.7	73	16.4	55	12.4	5	253	15.7
450	11.5	24	13.1	29	16.4	67	16.7	74	15.8	54	16.1	9	256	15.5
500	11.7	24	13.4	29	16.2	67	16.3	74	16.1	54	15.9	8	256	15.5
550	11.6	23	12.8	29	16.5	67	16.3	73	15.9	55	17.0	8	255	15.5
600	11.4	24	13.4	29	16.3	67	16.5	73	15.8	52	15.3	8	252	15.4
650	12.0	24	13.2	29	16.1	68	15.8	73	15.7	54	15.6	6	253	15.2
700	11.7	24	13.8	29	16.1	66	15.9	74	15.3	55	15.5	7	254	15.2

**Table 6.9:** Relation between bunker price, weighted average speed per vessel class and vessel deployment for each class. Weighted Average speed (W. Av. S.) is a weighted by the number of vessels deployed in the class (#v). The results are an average of five runs.



**Figure 6.6:** The weighted average speed (W.Av.S.), of an instance, the cargo transported in percentage of total available (Trnsp.), and the fleet capacity deployed in percentage of total volume, (Depl.) as a function of bunker price. The red dashed trend lines are based on a linear regression fit. The results are an average of five runs.

## 6.5 Conclusions

We have presented the competitive liner shipping network design problem where we include level of service requirements in the form of tight transit time restrictions on all demands while maintaining the ability to transship between services. To improve the networks, getting more realistic transit times and a better fleet utilization, we propose a method that can handle variable speed on all sailing legs in the network.

The proposed matheuristic can handle tight transit time restrictions on all

demands and adjust speed on all sailing legs. The core components of the matheuristic is an integer program considering a set of removals and insertions to a service and an integer program that adjust the speed of each service iteratively. We extend the integer program to consider how removals and insertions influence the transit time of the existing cargo flow on the service. Each iteration of the matheuristic provides a set of moves for the current set of services and fleet deployment along with a proposed sailing speed on each service leg, which lead to a potential improvement in the overall profit. The evaluation of the cargo flow for a set of moves requires solving a time constrained multi-commodity flow problem using column generation.

Extensive computational tests, including a sensitivity analysis on bunker price, show that the algorithm is applicable in practice and that it is possible to generate profitable networks for the majority of the instances in *Liner-lib* while considering level of service requirements. Especially for the larger instances the approach generates networks of good quality where the fleet is well utilized and the majority of demands are transported while satisfying transit time restrictions. Still, some smaller demands are not served and the fleet is not utilized completely, suggesting that further algorithmic improvements may lead to even better solutions. We expect that especially more flexibility in terms of possible vessel class swaps could improve the algorithmic performance and the quality of the generated networks.

## Acknowledgements

This project was supported by The Danish Maritime Fund under the Competitive Liner Shipping Network Design project. The authors would like to thank Guy Desaulniers for his contribution to the previous works from which this article was extended and to Alessio Trivella and Niels-Christian Fink Bagger for comments which helped improving the manuscript.

## 6.6 Appendix

In the following we introduce a mathematical formulation of the CLSNPD. This is partly based on Brouer et al. (2015) and extends the problem description of the LSNPD presented in Brouer et al. (2014a) to handle transit times and variable speed. The model enforces a weekly frequency resulting in a weekly planning horizon.

A solution to the CLSNPD is a subset of the set of all feasible services  $S$ . A feasible service consists of a set of ports  $P' \subseteq P$ , a number of vessels, and a vector of sailing speeds corresponding to each sailing leg such that the total round trip time is a multiple of a week. A weekly frequency of port calls is obtained by deploying multiple vessels to a service. Let  $e(s) \in E$  be the vessel class assigned to a service  $s$  and  $n_{e(s)}$  the number of vessels of class  $e(s)$  required to maintain a weekly frequency. A round trip may last several weeks but due to the weekly frequency exactly one round trip is performed every week. The service time  $T_s$  is the time needed to complete the cyclic route.

An instance of the CLSNPD consists of the set of ports,  $P$ , with an associated port call cost  $c_p^e$  for vessels of class  $e(s)$ , (un)load cost  $c_U^p, c_L^p$ , transshipment cost  $c_T^p$  and berthing time  $B_p$  spent on a port call. Furthermore, we have a set of demands,  $K$ , available for transport each week where each demand has an origin  $O_k \in P$ , a destination  $D_k \in P$ , a quantity,  $q_k$ , a revenue per unit,  $z_k$ , a reject penalty per unit  $\tilde{z}_k$  and a maximal transit time,  $t_k$ . To service the routes, there is a set of vessel classes,  $E$ , with specifications for the weekly charter rate,  $C_e$ , capacity  $U_e$ , minimum ( $v_{min}^e$ ) and maximum ( $v_{max}^e$ ) speed limits in knots per hour, bunker consumption as a function of the speed,  $g_v^e$ , and bunker consumption per hour, when the vessel is idle at ports  $h^e$ . There are  $N_e$  vessels available of class  $e \in E$ . The price for one metric ton of bunker is denoted  $c_B$ . Finally we have a matrix,  $D$ , of the direct distances  $d_{ij}^e$  between all pairs of ports  $i, j \in P$  and for all vessel classes  $e \in E$ . The distance may depend on the vessel class draft as the Panama Canal is draft restricted. Along with  $d_{ij}^e$  follows an indication of the cost  $l_{ij}^e$  associated with a possible traversal of a canal.

The mathematical model of the CLSNPD relies on a set of service variables and a path flow formulation of the underlying time constrained multi-commodity flow problem as described in Karsten et al. (2015a).

We define a directed graph,  $G(V, A)$ , with vertices  $V$  corresponding to ports and arcs  $A$ . The set of arcs in the graph can be divided into (un)load arcs, transshipment arcs, sailing arcs, and forfeited arcs to reject demand. We as-

sociate with each arc  $a \in A$  a cost  $c_a$ , traversal time  $t_a$ , sailing speed  $v_a$ , and capacity  $C_a$ . The arcs used by service  $s$  is denoted  $A_s$ .

Let  $\Omega_k$  be the set of all feasible paths for commodity  $k \in K$  including forfeiting the cargo. Let  $\Omega(a)$  be the set of all paths using arc  $a \in A$ . The cost of a path  $\rho$  is denoted as  $c_\rho$  and it includes the revenue obtained by transporting one unit of commodity  $k$  sent along path  $\rho \in \Omega_k$ . The real variable  $x_\rho$  denotes the amount of commodity  $k$  sent along the path. The weekly cost of a service is  $c_s = n_{e(s)}C_{e(s)} + \sum_{(i,j) \in A_s} \left( c_B(h^{e(s)}B_p + g_{v(s)}^{e(s)}d_{ij}^{e(s)}) + c_j^{e(s)} + l_{ij}^{e(s)} \right)$  accounting for fixed cost of deploying the vessels and the variable cost in terms of the bunker and port call cost of one round trip. Define binary service variables  $y_s$  indicating the inclusion of service  $s \in S$  in the solution.

Then the mathematical model of the CLSNDP can be formulated as follows:

$$\min \quad \sum_{s \in S} c_s y_s + \sum_{k \in K} \sum_{\rho \in \Omega_k} c_\rho x_\rho \quad (6.23)$$

$$\text{s.t.} \quad \sum_{\rho \in \Omega_k} x_\rho = q_k \quad k \in K \quad (6.24)$$

$$\sum_{\rho \in \Omega(a)} x_\rho \leq U_{e(s)} y_s \quad s \in S, a \in A_s \quad (6.25)$$

$$\sum_{s \in S: e(s)=e} n_{e(s)} y_s \leq N_e \quad e \in E \quad (6.26)$$

$$x_\rho \in \mathbb{R}^+ \quad \rho \in \Omega_k, k \in K \quad (6.27)$$

$$y_s \in \{0, 1\} \quad s \in S \quad (6.28)$$

The objective (6.23) minimizes cumulative service and cargo transportation cost. As the cargo transportation cost includes the revenue of transporting the cargo, this is equivalent to maximizing profit. *The cargo flow constraints* (6.24) along with non-negativity constraints (6.27) ensure that all cargo is either transported or forfeited. *The capacity constraints* (6.25) link the cargo paths with the service capacity installed in the transportation network. *The fleet availability constraints* (6.26) ensure that the selected services can be operated by the available fleet. Finally, constraints (6.27) and (6.28) define the variable domains.

## Bibliography

- Agarwal, R. and Ergun, O. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196.
- Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11(2):186–208.
- Álvarez, J. F. (2012). Mathematical expressions for level of service optimization in liner shipping. *Journal of the Operational Research Society*, 63(6):709–714.
- Brouer, B., Álvarez, J., Plum, C., Pisinger, D., and Sigurd, M. (2014a). A base integer programming model and benchmark suite for liner shipping network design. *Transportation Science*, 48(2):281–312.
- Brouer, B., Desaulniers, G., Karsten, C., and Pisinger, D. (2015). A matheuristic for the liner shipping network design problem with transit time restrictions. In Corman, F., Voß, S., and Negenborn, R., editors, *Computational Logistics*, volume 9335 of *Lecture Notes in Computer Science*, pages 195–208. Springer International Publishing.
- Brouer, B., Desaulniers, G., and Pisinger, D. (2014b). A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59.
- Cheaitou, A. and Cariou, P. (2012). Liner shipping service optimisation with reefer containers capacity: an application to northern europe–south america trade. *Maritime Policy & Management*, 39(6):589–602.
- Christiansen, M. and Fagerholt, K. (2011). Some thoughts on research directions for the future: Introduction to the special issue in maritime transportation. *INFOR*, 49(2):75–77.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Fagerholt, K., Laporte, G., and Norstad, I. (2009). Reducing fuel emissions by optimizing speed on shipping routes. *The Journal of the Operational Research Society*, 61(3):523–529.
- Gelareh, S. and Meng, Q. (2010). A novel modeling approach for the fleet deployment problem within a short-term planning horizon. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):76–89.



- Hvattum, L. M., Norstad, I., Fagerholt, K., and Laporte, G. (2013). Analysis of an exact algorithm for the vessel speed optimization problem. *Networks*, 62(2):132–135.
- I. Norstad, K. F. and Laporte, G. (2011). Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19(5):853–865.
- Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015a). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138.
- Karsten, C. V., Ropke, S., and Pisinger, D. (2015b). Simultaneous optimization of container ship sailing speed and container routing with transit time restrictions. Technical report, Technical University of Denmark.
- Liu, Z., Meng, Q., Wang, S., and Sun, Z. (2014). Global intermodal liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 61:28–39.
- Meng, Q. and Wang, S. (2011). Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):695–708.
- Meng, Q., Wang, S., Andersson, H., and Thun, K. (2014). Containership routing and scheduling in liner shipping: Overview and future research directions. *Transportation Science*, 48(2):265–280.
- Notteboom, T. E. and Vernimmen, B. (2009). The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337.
- Plum, C., Pisinger, D., and Sigurd, M. M. (2014). A service flow model for the liner shipping network design problem. *European Journal of Operational Research*, 235(2):378–386.
- Psaraftis, H. N. (2015). *Green Transportation Logistics: The Quest for Win-Win Solutions*, volume 226. Springer.
- Psaraftis, H. N. and Kontovas, C. A. (2013). Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C: Emerging Technologies*, 26:331–351.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374.

- Reinhardt, L. B., Pisinger, D., Plum, C. E., Sigurd, M. M., and Vial, G. T. (2015). The liner shipping berth scheduling problem with transit times. *Transportation Research Part E: Logistics and Transportation Review*, page submitted.
- Ronen, D. (2011). The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society*, 62(1):211–216.
- Wang, S. and Meng, Q. (2012a). Liner ship fleet deployment with container transshipment operations. *Transportation Research Part E: Logistics and Transportation Review*, 48(2):470–484.
- Wang, S. and Meng, Q. (2012b). Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B: Methodological*, 46(5):615–633.
- Wang, S. and Meng, Q. (2012c). Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):701–714.
- Wang, S. and Meng, Q. (2014). Liner shipping network design with deadlines. *Computers and Operations Research*, 41(1):140–149.
- Zacharioudakis, P. G., Iordanis, S., Lyridis, D. V., and Psaraftis, H. N. (2011). Liner shipping cycle cost modelling, fleet deployment optimization and what-if analysis. *Maritime Economics & Logistics*, 13(3):278–297.



## Part III

# Service Selection with Limited Transshipments



# Container Shipping Service Selection and Cargo Routing with Limited Transshipments

with Anantaram Balakrishnan<sup>1</sup>

---

## Abstract

We address the tactical planning problem facing container liner shipping companies of selecting a set of sailing services from a given pool of candidate services and routing demand over the chosen services so as to maximize profit. One of the distinctive features of our model is that it incorporates limits on the number of transshipments for each container, a common requirement in practice. We propose a new stage-indexed multi-commodity arc flow model that is based on an augmented network containing arcs (representing sub-paths) between every pair of ports visited by each candidate service. This sub-path construct permits us to accurately model transshipment costs and enforce transshipment limits. To accelerate the solution procedure, we outline a preprocessing procedure that exploits the routing requirements to reduce problem size, develop valid inequalities to strengthen the linear programming relaxation, and propose an optimization-based heuristic algorithm. We present computational results for realistic problem instances from a benchmark suite of liner shipping problems.

---

<sup>1</sup>Balakrishnan, A. and Karsten, C. V. (2015). Container Shipping Service Selection and Cargo Routing with Limited Transshipments. Submitted to *Annals of Operations Research*

## 7.1 Introduction

Maritime transportation is a vital component of the modern global trading system. The share of goods transported globally on container ships has grown steadily over the past decades, and is expected to increase further due to the economic and environmental advantages of ocean transport compared to other modes. Since container ships are very expensive to acquire and operate, liner shipping companies need to utilize their assets effectively by judiciously choosing their sailing routes and deciding which demands to meet. The goal is to maximize profit while ensuring adequate service to customers. We address this problem by developing an optimization model that permits container ship operators to select the best set of services (sailing routes with associated fleet assignment and service frequency) to operate. An important feature of our model is its ability to incorporate limits on the number of transshipments, a common practice among liner shipping companies to assure good service to customers (Brouer et al., 2014a). To meet these requirements, we restrict the maximum permissible transshipments for each container. These limits are analogous to the hop constraints introduced by Balakrishnan and Altinkemer (1992). We refer to the tactical planning problem of selecting an appropriate set of shipping services as the Liner Service Planning (LSP) problem which is defined as follows. Given the anticipated demand between various ports that the company serves, the problem entails selecting a subset of services, defined as cyclic (possibly non-simple) sailing routes with associated assignment of shipping fleet, from a given pool of candidate services and transporting as much demand as possible over the chosen services, subject to transshipment limits, so as to maximize net profit. We propose a novel multi-commodity model based on flows along sub-paths for each stage of a commodity's route to capture the transshipment constraints and accurately model transshipment costs; the model also readily incorporates practical container routing issues such as cabotage rules, regional policies, and embargoes. For this model, we outline a preprocessing procedure that exploits the routing requirements and our new augmented network structure to reduce problem size. We also propose valid inequalities to accelerate the solution procedure, and develop an optimization-based heuristic procedure to generate good initial solutions. To demonstrate the effectiveness of our model solution procedure, we present computational results for realistic problem instances based on the *Liner-lib* benchmark problems. So, as we elaborate later, the paper provides new contributions along three dimensions - modeling, methodology, and application - that are all valued in the operations research and transportation literature.

### 7.1.1 Background

The International Maritime Organization (IMO, 2012) estimates that 90 % of global trade is carried by sea, with container ships transporting around 60 % of the seaborne goods by value. Modern cost and energy efficient container vessels can carry almost 20,000 twenty-foot equivalent units (TEU) of cargo. Since these ships cost more than 100 million dollars per vessel, operating a global ocean transportation network requires enormous capital investments. With such large investments, it is necessary to ensure high utilization of the ships to be able to offer low shipping cost which is the main advantage of ocean transport over other modes of transportation. Global liner shipping networks divide their geographical coverage regions into major trade lanes that follow the North-South, East-West, and intra-regional trade patterns of the world. Within each trade lane, a carrier may operate multiple services, each consisting of a cyclic sailing route that visits a given subset of ports (a port may be visited multiple times on the route). Each service consists of several vessels of approximately the same size that ply the route at roughly equally spaced intervals.

Although cost is an important factor for ocean cargo transportation the competitiveness of a carrier also depends on service assurances, such as limited transshipments, that the shipping line can provide (Brouer et al., 2014a). A good network will exploit transshipment opportunities at intermediate ports (that are visited by multiple services) to ensure high fleet utilization. However, it is also important for liner shipping companies to explicitly limit the number of transshipments of each container since customers prefer to have no more than two or three transshipments in order to reduce the risk of damage or loss, missed connections, and long layover times. Some types of cargo (e.g., hazardous goods or high value items) may need to be transported without any transshipments whereas others may not require tight limits on the number of transshipments. Therefore, the planning model must have the ability to impose demand-specific transshipment limits. Thus, liner shipping companies face trade-offs between utilization and level of service when planning their services. Container routing decisions may also be constrained by other operational policies and restrictions that we discuss in the later sections. Brouer et al. (2014a) provide a broader introduction to the domain of liner shipping.

### 7.1.2 Planning Problems in Liner Shipping

Liner shipping entails decisions at the strategic, tactical, and operational levels. The various planning problems in this context are summarized in Agarwal and



Ergun (2008) and more recently by Kjeldsen (2011) who discuss a classification of routing and scheduling problems in liner shipping. Meng et al. (2014) review the literature on models that address different levels of planning. The overall decisions at the strategic level in liner shipping focus on the size and mix of the operating fleet and the general configuration of the network, decisions that deal with the acquisition and deployment of highly capital intensive assets. At the tactical level, a carrier must determine which services to operate, the vessels used for these services, and the offered schedule. Once a network configuration has been decided, carriers usually make periodic adjustments when adding new services or modifying existing routes. At the operational level, a carrier must decide which demands to meet and how to route these demands; these decisions are referred to as the cargo routing problem. Other short-term operational decisions include responding to disruptions and repositioning empty containers. A significant portion of the costs for operating a network are determined by the long and medium term planning decisions, whereas the revenues depend on the short term cargo selection and routing choices. So, liner shipping networks are often said to have a two-tier cost structure, one concerning asset costs that depend on fleet and network design and the other consisting of operational costs and revenues that depend on cargo routing (Agarwal and Ergun, 2008; Álvarez, 2009).

A challenging part of the planning process is the ripple effect on traffic throughout the network when introducing new services in one portion of the network. Generally, companies in the liner shipping industry rely on experienced planners to manually design candidate routes and associated services that are consistent with the company strategy. To help network planners adapt to the ever changing market and operational conditions, we propose a decision support model to select the best subset from these candidate services that the company can operate using available ships so as to maximize profits. Our model is appealing to practitioners because it permits planners to specify which services to consider; these candidate services may include those that have previously been part of the network as well as newly designed services. With manual planning, liner shipping companies usually optimize one trade lane at a time or services within a single region. Using our proposed model, planners can consider the interactions between flows on services throughout the network, thereby making more effective overall service choices. We illustrate the application of our model and solution method for test problems based on realistic networks and routes with varying characteristics. Our tactical planning model is different from and complements the existing research literature on liner shipping which has largely focused either on network design or on lower level planning problems such as fleet deployment, sailing speed optimization, bunker optimization, and scheduling.

The main scientific contributions of this paper are as follows: we (i) present a new model for tactical planning of containership operations for liner companies, (ii) propose modeling and methodological improvements to solve the problem effectively, and (iii) apply the approach to realistic shipping networks to demonstrate its effectiveness. Specifically, we frame a tactical planning problem that has not been previously addressed, namely, the general problem of selecting an optimal set of services from a given candidate set and routing containers on the chosen services. A distinctive feature of our approach is its consideration of variable transshipment limits governing routing decisions that have not been considered in prior network design models. For this problem, we propose a novel network construction and multi-commodity flow model based on flow along sub-paths, indexed by transshipment stage, to accurately capture container transshipments and costs. This model is new to the literature. From a theoretical point of view, we show that the problem is NP-hard. To effectively solve this problem, we propose preprocessing procedures to reduce the model size, valid inequalities to tighten the model's linear programming relaxation, and an optimization-based heuristic procedure to generate good initial solutions. These enhancements reduce the overall computational time to solve the problem. Finally, we test our model and methodology on problem instances that are based on realistic networks; our computational results demonstrate that our solution method is effective and the model yields practical solutions.

The rest of this paper is organized as follows: Section 7.2 reviews the literature on related problems. Section 7.3 formally defines the LSP problem and introduces the model as well as some improvements. Section 7.4 discusses algorithmic aspects of the implementation, presents computational results, and discusses the use of the model. Section 7.5 contains concluding remarks.

## 7.2 Literature Review

Christiansen et al. (2004, 2013) provide comprehensive reviews of the literature published during the past few decades on maritime transportation. Recently, much of the focus in this stream of literature has been on liner shipping due to the importance of this context and the challenges in solving these problems. Researchers have addressed problems at all three levels – strategic, tactical, and operational levels. We outline some of the literature on service selection and network design that is related to our work.

### 7.2.1 Liner Shipping Network Design

Meng et al. (2014) provide a comprehensive review of mathematical optimization methods for strategic network design, concluding that no approach has been successful in designing networks of realistic size while also accounting for industry constraints, service requirements, and routing decisions. Most papers employ heuristic methods, and the exact methods proposed so far are only able to solve small problem instances. Agarwal and Ergun (2008) consider a simultaneous ship scheduling and cargo routing model. The model creates cyclic routings for a set of vessel classes with a rough schedule and a weekly frequency constraint. They introduce a time-space graph spanning each weekday to capture the scheduling decisions, but do not consider cargo transit times. The model allows for any number of transshipments and ignores the cost of these transshipments. The authors' algorithm, based on Benders decomposition and column generation, scales relatively well. Álvarez (2009) considers the problem of joint routing and deployment of a fleet of container vessels to generate cyclic routes. No frequency restrictions are imposed, and cargo travel times and schedules are not considered. An unlimited number of transshipments are permitted at a cost (not captured accurately). The paper solves the model using a standard MIP solver and a tabu-search method. Brouer et al. (2014a) extend the model of Álvarez (2009) to correctly account for transshipment costs (but with no limits on transshipments) and introduce a frequency restriction. A heuristic column generation method is able to produce networks of reasonable size but with varying quality. Using the model presented in Brouer et al. (2014a), Brouer et al. (2014b) present a matheuristic to perform incremental network optimization taking the cargo flows into consideration. They also present a heuristic for designing services from scratch. The solutions are promising for larger instances, but do not consider service level issues such as transit time or transshipment restrictions. Brouer et al. (2015) extend the model to incorporate transit time, but allow unlimited transshipments. The transit time restrictions do not degrade algorithmic performance significantly since these restrictions reduce the solution space for the cargo routing sub-problem. Gelareh et al. (2010) study the network design problems facing two competing liner shipping companies on a hub-and-spoke network where the market share depends on transit time and transportation cost. Reinhardt and Pisinger (2012) propose an exact branch-and-cut algorithm for the container shipping network design problem. Their model allows transshipments and generates non-simple cyclic routes, but does not limit either transit time or number of transshipments and does not impose any frequency requirements on the sailing routes. The proposed method can only solve fairly small problems. Wang and Meng (2014) present a non-linear mixed integer programming model for network design, taking into account transit time but they do not per-

mit transshipments, and propose a column generation-based heuristic for this model. Computational results are presented for a Europe-Asia network with 12 ports. Meng and Wang (2011) consider the problem of selecting services over a set of hub and feeder ports. Their model assumes that containers originating in a feeder port cannot be transshipped more than two times, and each feeder port is assigned to exactly one hub port, which is the only location at which a container can transfer to another service. Further, cargo between two hubs cannot be transshipped. With these assumptions, the possible container paths are quite limited and can easily be enumerated. The objective is to minimize cost while ensuring that all ports and flows are served (revenue is not considered). The authors describe a mixed-integer programming model, and present computational results for an Asia-Europe-Oceania network containing six hubs at which transshipments occur. Finally, Plum et al. (2014) propose a branch-and-cut-and-price algorithm to design a single service that can visit up to 25 ports.

### 7.2.2 Related Problems

In another stream of literature, several authors have studied restricted problems for tactical and operational decisions (see Christiansen et al. (2004, 2013)). These models often require additional assumptions to make the models solvable. As an example, Wang and Meng (2011) use pre-defined container paths and study container routing and schedule design. Their model adds a penalty (or bonus) cost for longer (shorter) transit times and minimizes the transshipment cost. At the tactical level, Wang and Meng (2012) study schedule design and speed optimization under uncertainty. Meng and Wang (2012) study the fleet deployment problem in a space-time network where they consider transit times. Wang et al. (2013) study time-constrained container routing in a restricted network for paths between one origin-destination pair. They allow at most one transshipment, and assume that the transshipment cost is the same at all ports. Brouer et al. (2013) study the operational level problem of recovering schedules.

## 7.3 LSP Problem: Definition and Model Formulation

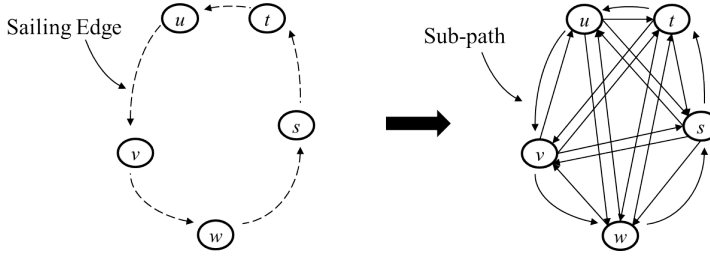
In this section, we formally define the LSP problem, present a multi-commodity flow model defined over our augmented network, and describe preprocessing

methods to reduce problem size while also incorporating practical routing constraints. We then present a set of valid inequalities to strengthen the model. We formally introduce the problem next, introduce some notation, and describe an appropriate network topology to model the problem.

### 7.3.1 Problem Statement and Network Representation

Given a pool of candidate services (i.e., specified sailing routes, each with an associated fleet assignment and service frequency) and the estimated demand between various ports, we wish to select the best subset of services from the candidate pool, determine how much of each origin-to-destination demand to serve, and route these demands on the chosen services so as to maximize net profit, which is revenue from served demand less costs of deploying and operating ships and for container loading, unloading, and transshipments. The routing decisions must satisfy limits on the number of permitted transshipments for each demand; further, the ships needed to operate the chosen services must not exceed fleet availability, and their container loads on each sailing leg must be no more than the ship capacity.

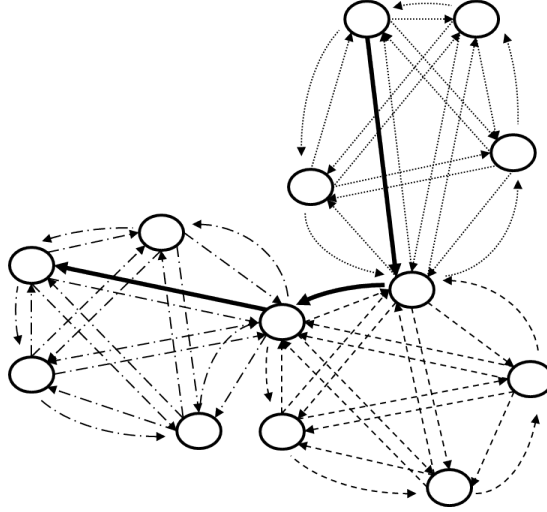
The liner shipping transportation network consists of a set of ports  $p \in P$ , a set of candidate services  $r \in R$ , and the sailing edges  $e \in E^r$  in each service  $r$ , representing the portion of a ship's itinerary between two successive ports of call on the service route. The number and types of ships needed for service  $r$ , the capacity (in terms of number of containers) of each sailing edge of that service  $t_e$ , and the cost  $f_r$  of selecting that service are given. In practice, the available capacity of a sailing leg may differ from the capacity of the vessel class assigned to the corresponding service, e.g. if there is a vessel sharing agreement (VSA) for the service. Therefore, rather than assume the same capacity on each sailing edge of a service, we permit the capacity to vary by edge. The cost  $f_r$  of selecting service  $r$  includes the amortized cost of the vessels assigned to the service, fuel and other operating costs for sailing and idling, canal costs, and port call costs. As inputs to the model, we are given the container traffic available to be transported between various origin-destination ports. We associate a commodity  $k \in K$  with each such demand; this commodity originates at port  $s_k$  and needs to be transported to destination port  $t_k$ . For each commodity  $k$ , let  $d_k$  denote the forecasted number of containers available for transport. We are not required to transport all of this demand, but we consider a cost or penalty of  $q_k$  per container for not fully meeting demand for commodity  $k$ . This cost may represent, for instance, the opportunity loss for unmet demand; in this case,  $q_k$  is the revenue per container for commodity  $k$ . We permit splitting demands, i.e., we can route containers of a commodity along multiple paths from origin to destination.



**Figure 7.1:** Sub-paths corresponding to a service that visits 5 ports.

A common way to model the flows of commodities on liner networks is to define commodity flow variables for each arc of the network, i.e., the number of containers carried on each sailing edge of every service. However, capturing limits on transshipments with such a model is very difficult (it requires defining many additional binary variables to model inter-service transfers, and is particularly cumbersome and impractical when demands can be split among multiple routes). Instead, one of the distinctive and key features of our model is the way we define the underlying network for multi-commodity flows. Specifically, to incorporate limits on the number of transshipments and capture their costs, we introduce an augmented multi-commodity flow network based on sub-paths. We define a sub-path as the portion of a ship's service between any two ports of call (not necessarily consecutive) on the corresponding route. So, if a service  $r$  visits  $n_r$  ports, we can have as many as  $n_r(n_r - 1)$  associated sub-paths on that service. But, as we note later, operational policies may preclude using some of these sub-paths. Our augmented network contains one node for each port and one arc for each sub-path of every service. Figure 7.1 shows an original service covering five ports, and the corresponding network with sub-paths. Figure 7.2 shows the augmented graph for a liner shipping network with three services, spanning 13 ports. As Figure 7.1 illustrates, each sub-path is composed of a sequence of underlying or embedded sailing edges on the service. The cost of this sub-path, therefore, includes the transportation costs for the embedded sailing edges.

In our model formulation, the flow of a commodity on a sub-path from port  $i$  to port  $j$  will represent the containers that are loaded on the service corresponding to the sub-path at port  $i$  (possibly arriving at that port on a different service) and unloaded (and possibly transshipped) at port  $j$ . We denote this sub-path as  $\langle i, j \rangle$ . The unit cost (per container) of this sub-path includes the variable costs of the sailing edges on this sub-path as well as the costs of loading, unloading, and transshipments at the starting and ending ports. We distinguish between the cost of transshipping from one service to another at an intermediate port



**Figure 7.2:** Augmented network for an example with 3 services spanning 13 ports.

from the cost of loading/unloading at the origin/destination of the commodity. In practice, the cost of a transshipment (unloading and re-loading) is around 15 % less than the total cost of one unload and one load operation (some ports have a larger cost difference). Our model permits distinguishing these costs by making the sub-path costs commodity-dependent. So, for a sub-path from port  $i$  to port  $j$ , if port  $i$  (port  $j$ ) is the origin (destination) of commodity  $k$ , then this commodity incurs only the appropriate loading (unloading) cost at that port; otherwise, if port  $i$  or  $j$  is a transshipment port for commodity  $k$ , the relevant cost is the cost of transferring the container from one service to another. The sub-path construct is not limited to simple routes that visit each port exactly once; rather, it also extends to complex routes such as a butterfly or conveyor belt services Brouer et al. (2014a) that may visit a port more than once. We refer to such ports as multi-visit ports. In our model, we include sub-paths that go through a multi-visit port  $l$  more than once. Instead of assigning containers to such a sub-path from, say, port  $i$  to port  $j$ , the solution may alternatively use two shorter sub-paths, one from  $i$  to  $l$  and the other from  $l$  to  $j$ ; in this case, containers must be first unloaded and then re-loaded at port  $l$ . The model's decision on whether to use the single longer sub-path or the two shorter sub-paths depends on the cost tradeoffs (i.e., cost of additional sailing edges in the longer sub-path versus additional load/unload costs for the two sub-paths) and capacity usage on the sailing edges of the longer sub-path.

In the augmented network, the number of transshipments for a container is one less than the number of sub-paths on which the commodity flows. Hence, we can enforce the transshipment limits by limiting the number of sub-paths or hops on a container's route. Balakrishnan and Altinkemer (1992) were among the first researchers to study hop-constrained problems in the context of network design. The use of sub-paths as the basis for our augmented network is novel, although it has some similarity to the segment construction introduced by Meng and Wang (2011) and the links defined by Bell et al. (2011). In these other papers, the segments and links are simpler or not service-specific, whereas in our model each sub-path has an associated service, providing more modeling flexibility.

### 7.3.2 Model Formulation

In our multi-commodity model defined over the augmented network, instead of adding explicit constraints on the number of transshipments, we define flow variables for each possible transshipment stage of every commodity. We index these stages consecutively from 1 to the maximum number of permitted transshipments for a commodity. In this scheme, stage  $h$  represents the  $h^{th}$  service (in sequence) that a container is routed on since it departed from its origin. These stage-indexed flow variables are then linked across stages using appropriate flow conservation constraints. Thus, our model implicitly accounts for the transshipment limits by permitting, for each commodity, only as many stages as the transshipment limit for that commodity. Before presenting our model formulation for the LSP problem, we introduce some additional necessary notation in Tables 7.1 and 7.2.

As noted above, for each commodity  $k$ , we define flow variables for every sub-path  $\langle i, j \rangle$  that the commodity can use as the  $h^{th}$  stage. We also need variables to represent the amount of unmet demand for each commodity, and binary variables to decide which among the candidate services to select. The definitions of these decision variables are stated in Table 7.3.



Set	Definition
$P$	Set of all ports in the shipping network; $i, j, \dots \in P$
$K$	Set of commodities, $k \in K$
$R$	Set of candidate services, $r \in R$
$E^r$	Set of sailing edges in service $r \in R$ ; $e \in E^r$
$A^r$	Set of sub-paths in service $r$ , $\langle i, j \rangle \in A^r$ is the sub-path of service $r$ from port $i$ to port $j$
$B_r^{hk}$	Set of sub-paths $\langle i, j \rangle \in A^r$ of service $r$ eligible as the $h^{th}$ transportation stage for commodity $k$
$P^{hk}$	Set of intermediate ports (excl. destination $t_k$ ) that can be reached in stage $h$ for commodity $k$
$V$	Set of available vessel types, $v \in V$

**Table 7.1:** Overview of sets.

Parameter	Definition
$f_r$	Cost of service $r$
$c_{ijr}^{hk}$	Cost per container if commodity $k$ is routed on sub-path $\langle i, j \rangle$ on service $r \in R$ in the $h^{th}$ stage
$d_k$	Forecasted number of containers available to be transported, for commodity $k$
$h_k$	Maximum number of permitted transshipments (stages) for commodity $k$ .
$m_r^v$	Number of vessels of type $v$ needed for service $r$
$n^v$	Number of available vessels of type $v$
$q_k$	Penalty for not meeting one unit (container) of demand for commodity $k$
$t_e$	Capacity (number of containers) of sailing edge $e^r$ of service $r$

**Table 7.2:** Overview of parameters.

Variable	Definition
$u_{ijr}^{hk}$	Flow of commodity $k$ on sub-path $\langle i, j \rangle$ of service $r$ as the $h^{th}$ stage, for $h = 1, 2, \dots, h_k$
$z_k$	Amount of demand (containers) not met for commodity $k \in K$
$x_r$	1 if service $r$ is used, 0 otherwise, for all $r \in R$

**Table 7.3:** Overview of decision variables.

We can now formulate the LSP problem as the following mixed-integer program:

$$\min \quad \sum_{r \in R} f_r x_r + \sum_{k \in K} \sum_{r \in R} \sum_{\langle i, j \rangle \in A^r} \sum_{h=1}^{h_k} c_{ijr}^{hk} u_{ijr}^{hk} + \sum_{k \in K} q_k z_k \quad (7.1)$$

subject to

$$\sum_{r \in R} \sum_{\langle i, j \rangle \in B_r^{1k}} u_{ijr}^{1k} + z_k = d_k \quad k \in K \quad (7.2)$$

$$\sum_{r \in R} \sum_{i: \langle i, j \rangle \in B_r^{hk}} u_{ijr}^{hk} - \sum_{r \in R} \sum_{l: \langle j, l \rangle \in B_r^{h+1, k}} u_{jlr}^{h+1, k} = 0 \quad k \in K, j \in P^{hk},$$

$$h = 1, \dots, h_k - 1 \quad (7.3)$$

$$\sum_{k \in K} \sum_{h=1}^{h_k} \sum_{\langle i, j \rangle \in A^r(\langle i, j \rangle)} u_{ijr}^{hk} \leq t_e x_r \quad e \in E^r, r \in R \quad (7.4)$$

$$\sum_{r \in R} m_r^v x_r \leq n^v \quad v \in V \quad (7.5)$$

$$u_{ijr}^{hk} \geq 0 \quad k \in K, r \in R, \langle i, j \rangle \in B_r^{hk}, h = 1, \dots, h_k \quad (7.6)$$

$$z_k \geq 0 \quad k \in K \quad (7.7)$$

$$x_r \in \{0, 1\} \quad r \in R \quad (7.8)$$

The objective function (7.1) minimizes the total cost, consisting of the fixed and operating costs for the selected services, the cost of transporting goods on each sub-path including the costs for loading/unloading and transshipment at the starting and ending ports of the sub-path, and the penalty for unmet demand. Constraints (7.2) assign the flow of each commodity  $k$  to the sub-paths incident from the origin port for this commodity (in stage  $h = 1$ ), and specify that the total flow on these sub-paths together with the unmet demand (variable  $z_k$ ) must equal the commodity's demand. Constraints (7.3) are the flow balance constraints for intermediate stages ( $h > 1$ ). They require the total flow of a commodity entering a port  $j$  at stage  $h$  to equal the total flow leaving that port for stage  $(h + 1)$ . These constraints, over all the stages, together ensure that all flows of a commodity travel from origin to destination within the specified number of transshipments. For each commodity, we define flow variables (and include appropriate constraints (7.2) and (7.3)) so as to only allow flow on paths that use less than  $h_k$  sub-paths or stages. At the same time, the model permits splitting flow on multiple origin-to-destination paths. Constraints (7.4) serve to both impose the capacity of each sailing edge and also ensure that we assign flows to a sub-path only if the corresponding service  $r$  is selected. The left-hand side of these constraints include the total flow on

all sub-paths that use the sailing edge  $e$ ; these flows can be positive only if the corresponding service  $r$  is chosen (i.e.,  $x_r = 1$ ) and must not exceed the capacity of edge  $e$ . Constraints (7.5) specify that the total number of vessels of each type needed to operate the chosen services must not exceed the liner company's available fleet of that vessel type. Constraints (7.6) to (7.8) are the non-negativity and binary constraints.

By defining the flow variables  $u_{ijr}^{hk}$  only for the sub-paths  $\langle i, j \rangle$  in the set  $B_r^{hk}$ , we only consider the eligible sub-paths for routing a specific commodity  $k$ , thus significantly reducing the size of the model. For instance, only sub-paths adjacent to the origin port of a commodity are eligible as the first stage. The number of variables in the model depends on the number of commodities, number of services, number of sub-paths for each commodity (which is a quadratic function of the number of ports visited by the service), and the maximum number of transshipments permitted. The number of constraints is largely determined by the number of commodities, transshipment ports, and sailing edges. For a problem instance with a maximum transshipment limit of  $H$  each additional service  $r$  added to the service pool will introduce an additional number of variables and constraints which are bounded by  $|E^r|(|E^r| - 1)|K|H$  and  $|E^r|$ , respectively, assuming that adding a service does not add a new port of call for a commodity in constraints (7.3).

### 7.3.3 Problem Complexity

As the following proposition shows, the LSP problem is NP-hard, i.e., it is computationally intractable in a theoretical sense.

**Proposition:** The LSP problem is NP-hard

**Proof:** To establish this result, we transform the knapsack problem, which is NP-hard, to a special case of the LSP problem in polynomial time. In the knapsack problem we have  $n$  items available, and a subset of the items must be selected such that the total profit of the chosen items is maximized. Let  $\pi_i$  denote the profit and  $\lambda_i$  the weight of an available item  $i$ . An item can only be selected or not selected and there is a total weight budget of  $b$  available to select items. Given any instance of the knapsack problem, for each item  $i$ , we create a service visiting two ports, a port  $i$  that is only covered by this service and a common hub-port 0 that is visited by all services. For each item  $i = 1, \dots, n$ , we create a commodity with demand from port  $i$  to port 0, and set the cost of not meeting this demand (i.e., the coefficient of the  $z_k$  variable in our model) to  $\pi_i$ . The service visiting port  $i$  can meet all available demand at  $i$  and require  $\lambda_i$  vessels. We have  $b$  available vessels. The equivalent LSP

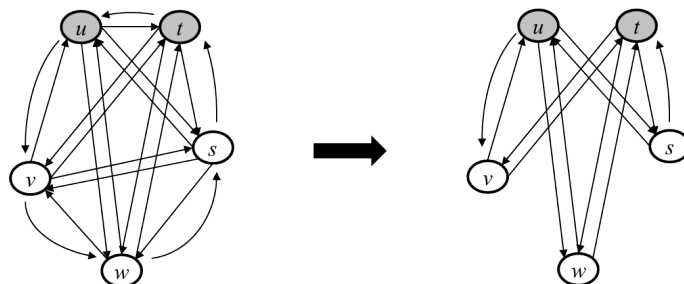
problem then entails selecting items (services) so as to minimize the cost of unmet demand (i.e. maximize profit) while satisfying the weight budget; we only get the profit  $\pi_i$  if service  $i$  is selected. Since the knapsack problem is NP-hard (Nemhauser and Wolsey, 1988), so is the LSP problem. ■

### 7.3.4 Incorporating Trade and Operational Policies

Next, we discuss some practical issues - trade policies, regulations, and operational policies - that our model can readily accommodate. Further, in some cases, these requirements reduce the number of variables, making the problem easier to solve.

#### 7.3.4.1 Maritime Cabotage Rules and Embargoes

Due to the trade policies of various countries, liner shipping companies must follow certain cabotage rules such as restrictions on the transshipment of goods in the destination country and internal transport of goods within a country. There are also rules for hazardous goods and for various embargoes between countries. In the standard network path flow formulation for cargo routing, one way to account for cabotage rules may be to incorporate them as resource constraints in the shortest path calculations for solving the routing sub-problems. For our model, on the other hand, we can simply remove sub-paths that are not permitted for a commodity during a preprocessing stage. Figure 7.3 shows an example with five ports, two of which are in one country (ports  $u$  and  $t$ , shaded in the figure) and the other three in another country. Then, for commodities



**Figure 7.3:** Eliminating sub-paths that violate maritime cabotage rules. The sailing direction is counterclockwise.

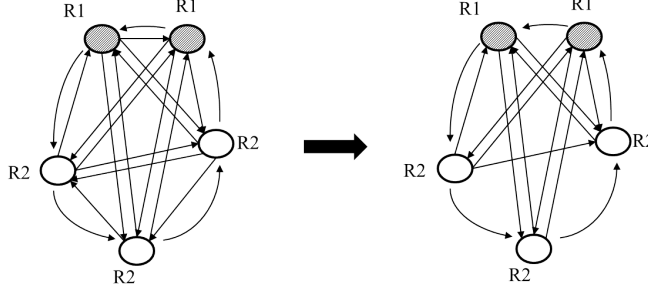
that originate in one country and are destined to the other, if the cabotage rules prohibit transshipment of containers within the originating or destination country, we can eliminate several sub-paths for such commodities, as shown in the right-hand side of the figure. To implement the model, we can delete any sub-paths that violate cabotage rules when building the sub-path graph, and omit the corresponding variables from the model, thereby reducing the size of  $B_r^{hk}$ . Therefore, incorporating the cabotage rules actually reduces the problem rather than complicating it, while also making the model more realistic. Other special rules such as embargoes have a similar structure and can be included as well.

#### 7.3.4.2 Operational Policies

Networks that are optimized to minimize cost and increase network utilization often assign containers to detours that have unused capacity (Karsten et al., 2015). In practice, even without limits on transit time or number of transshipments, there are many container routes that a planner would not consider to be appropriate. With our LSP model, we can readily incorporate operational policies that restrict the permissible flow paths for specific commodities. These restrictions not only lead to more realistic flows but also reduce the model size since the sub-paths violating these policies for a commodity can be removed directly from the set  $B_r^{hk}$ . To illustrate this feature, we discuss one practical routing policy based on flows across geographical regions.

#### 7.3.4.3 Regional Policy

This policy states that if the two ports at which a container is loaded and unloaded from a service lie in the same region, then it should not be routed via another region. To incorporate this policy, we remove sub-paths that start and end in the same region but include an intermediate port in another region. For instance, for a container going from Vancouver to Panama, we rule out paths that visit Asia since flow on such a sub-path would be undesirable. See Figure 7.4 for an illustration of this policy. In this example, ports  $u$  and  $t$  are in region R1 and the remaining ports are in region R2; so, applying the regional policy leads to the reduced network shown in the right-hand side of the figure. The procedure in Figure 7.5 implements this sub-path elimination rule.



**Figure 7.4:** Example of a regional policy: eliminate sub-paths that start and end in the same region but visit an intermediate port in another region. The sailing direction is counterclockwise.

```

for all  $r \in R$  do
  for all  $\langle i, j \rangle \in A^r$  do
    if sub-path  $\langle i, j \rangle$  starts and ends in Region 1 (Region 2) but includes
    a port in Region 2 (Region 1) then
      for all  $k \in K$  and  $h = 1, 2, \dots, h_k$  do omit variable  $u_{ijr}^{kh}$ 

```

**Figure 7.5:** Sub-path elimination rule.

#### 7.3.4.4 Other Policies

The commodity-specific sub-path variables and transshipment limits permit us to easily incorporate other routing policies such as those based on distance. For instance, containers with origin and destination that are relatively close may be allowed only one transshipment. Since our model permits commodity-dependent transshipment limits, we can implement this requirement by removing variables corresponding to more than one transshipment. We can also incorporate general inter-regional transshipment limits. And, for commodities with origin and destination in the same region, we can remove sub-paths that start or end in other regions. In the current formulation, although imposing limits on end-to-end transit time is not straightforward, by judiciously specifying the permitted sub-paths for a commodity we can ensure that, for each sub-path, there is at least one origin-to-destination path via this sub-path that satisfies the allowed transit time.

```

for all  $i \in P_k^+$  do
  if  $\nexists u_{ijr}^{2k}$  for any  $j \in P_k^-, r \in R$  then omit  $u_{s_k i r}^{1k} \forall r \in R$ 
for all  $j \in P_k^-$  do
  if  $\nexists u_{ijr}^{2k}$  for any  $i \in P_k^+, r \in R$  then omit  $u_{j t_k r}^{3k} \forall r \in R$ 

```

**Figure 7.6:** Variable elimination.

### 7.3.5 Problem Reduction

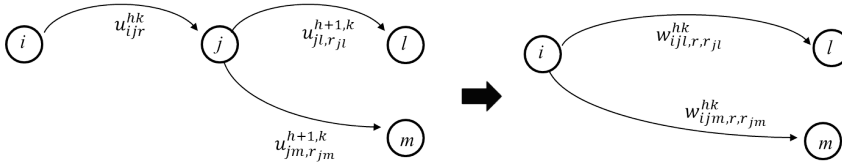
After incorporating the previous operational policies, several of the flow variables in the model for a commodity may correspond to sub-paths that do not belong to any origin-to-destination path satisfying the commodity's transshipment limit. In this situation, we can eliminate such variables and reduce the problem size. As an illustration, the following variable elimination procedure applies when a commodity  $k$  must have less than three transshipments.

Let  $P_k^+$  and  $P_k^-$  respectively denote the set of ports adjacent to the origin  $s_k$  and destination  $t_k$  of commodity  $k$ . Then, the procedure in Figure 7.6 eliminates several  $u$ -variables that cannot have positive values in any feasible solution. As another variable reduction strategy, we can merge consecutive sub-paths for a commodity when an intermediate port has only one predecessor or successor at a particular stage. Correspondingly, we replace the flow variables on the sub-paths incident at this port with composite variables. To illustrate this method, consider a port  $j \in P_k^{hk}$  that has only one sub-path  $\langle i, j \rangle$  entering this port as the  $h^{th}$  stage for a commodity  $k$ . Suppose  $r$  is the service corresponding to this sub-path. Let  $AO_j^{h+1,k}$  denote the set of all sub-paths  $\langle j, l \rangle$  from port  $j$  that can serve as the  $(h+1)^{st}$  stage for commodity  $k$ . For each of these sub-paths  $\langle j, l \rangle$ , let  $r_{jl}$  denote the corresponding service. In this configuration, we can replace the original flow variables on the inbound path  $\langle i, j \rangle$  and outbound paths  $\langle j, l \rangle$  with combined flow variables that bypass the intermediate node  $j$  and also delete the flow conservation constraint for the  $h^{th}$  stage at this node. Specifically, we perform the following local transformations at node  $j$  to get an equivalent model:

- for each outbound sub-path  $\langle j, l \rangle \in AO_j^{h+1,k}$  at node  $j$  corresponding to stage  $(h+1)$  for commodity  $k$ , define a "composite" flow variable  $w_{ijl, r, r_{jl}}^{hk}$  which denotes the flow of containers from node  $i$  to node  $l$  via node  $j$  as consecutive stages  $h$  and  $(h+1)$  for commodity  $k$ . Assign a cost (coefficient in the objective function (7.1)) of  $(c_{ijr}^{hk} + c_{jl, r_{jl}}^{h+1,k})$  to this variable;

- for each  $\langle j, l \rangle \in AO_j^{h+1,k}$ , replace the original flow variable  $u_{jl}^{h+1,k}$  with variable  $w_{ijl,r,r_{jl}}^{hk}$  in the constraints of the model formulation;
- replace the original flow variable  $u_{jlr}^{hk}$  with  $\sum_{\langle j,l \rangle \in AO_j^{h+1,k}} w_{ijl,r,r_{jl}}^{hk}$  in the constraints of the model formulation; and,
- delete the flow conservation constraint (7.3) at node  $j$  corresponding to commodity  $k$  and stage  $h$ ;

Figure 7.7 pictorially illustrates this transformation. In this figure, intermediate port  $j$  has only one incoming sub-path for commodity  $k$  at stage  $h$ , permitting us to merge this sub-path with the succeeding sub-paths.



**Figure 7.7:** Combining variables for a commodity  $k$  at a node  $j$  with in-degree of one.

With the above modifications, the resulting model is equivalent to the original model (i.e., for each feasible solution to one model, the other model has an equal cost feasible solution). An analogous transformation applies when there is only one sub-path  $\langle j, l \rangle$  leaving an intermediate port  $j$  at some stage  $h$ ; in this case, we can combine the appropriate incoming sub-path flow variables and the outgoing flow variables into composite variables. Moreover, this process extends to more complex situations (e.g., when a node  $j$  has only one incoming and one outgoing sub-path, in which case we can combine flow variables across three stages). These transformations reduce the size of the model by eliminating some variables and constraints.

### 7.3.6 Strengthening the Model

We next present a class of valid inequalities to strengthen the LSP model, i.e., to increase the value of its linear programming relaxation lower bound. For ease of exposition, we consider the basic model of Section 7.3.2, but the inequalities also extend to the model after applying the reduction methods of Sections 7.3.4 and 7.3.5. The inequalities we propose are essentially disaggregate forcing



constraints that combine the linkage between the flow and service selection variables and the capacities on individual sailing edges. The validity of the inequalities stem from the fact that, for a given commodity, the flow of this commodity on a sub-path cannot exceed the minimum of the demand and the capacity of that sub-path.

A commodity can flow on a sub-path only if the corresponding service is selected; further, the flow must not exceed the demand of the commodity or the capacity of every sailing leg in the sub-path. These observations imply the following set of forcing constraints are valid for the LSP model:

$$u_{ijr}^{hk} \leq \min(d_k, \min_{e \in \langle i, j \rangle} t_e) x_r \quad k \in K, r \in R, \langle i, j \rangle \in B_r^{hk}, h = 1, 2, \dots, h_k \quad (7.9)$$

*Disaggregate (commodity-edge) constraints*

We can strengthen this inequality by considering mutually exclusive flows on the original sailing edges. For a given commodity and service, we consider the flow on all sub-paths that have a common sailing edge (at any stage  $h$ ). This flow must not exceed the smaller of the commodity's demand and the capacity of the sailing edge. This property holds because, with positive arc costs, no optimal solution will route a commodity multiple times over the same sailing edge. So, when  $d_k < t_e$ , the following inequality is stronger than inequality (7.9):

$$\sum_{h=1}^{h_k} \sum_{\langle i, j \rangle \in A^r(e)} u_{ijr}^{hk} \leq \min(d_k, t_e) x_r \quad k \in K, r \in R, e \in E^r \quad (7.10)$$

*Disaggregate (commodity-service) constraints*

We can develop additional valid inequalities based on sub-paths associated with a service. The flow of a commodity  $k$  on a service  $r$  at stage  $h$  must not exceed  $d_k$  if the service is chosen, motivating the inequality:

$$\sum_{\langle i, j \rangle \in B_r^{hk}} u_{ijr}^{hk} \leq \min(d_k, \min_{e \in E^r} t_e) x_r \quad k \in K, r \in R, h = 2, \dots, h_k - 1 \quad (7.11)$$

For  $h = 2, \dots, h_k - 1$ , inequality (7.11) strengthens the LSP model; for  $h = 1$  and  $h = h_k$ , this inequality is implied by our previous valid inequality (7.9).

**Require:** a solution to the LP-relaxation of LSP

```

while  $\mathbf{x}$  is fractional do
    fix integer elements of  $\mathbf{x}$  to 0/1
    select smallest fractional element  $x_s$ 
    if  $x_s < \alpha$  then fix  $x_s$  to 0
    else pick the largest fractional element  $x_l$ 
        if feasible in terms of fleet availability then fix  $x_l$  to 1
        else fix  $x_l$  to 0
    resolve LP-relaxation

```

**Figure 7.8:** LP-based heuristic.

### 7.3.7 LP-based Heuristic Solution

To obtain good solutions to the LSP problem, we apply a rounding heuristic that iteratively rounds (up or down) fractional values for the service selection ( $x_r$ ) variables in the solution to the linear programming (LP) relaxation to the problem. At each iteration, we select the highest or lowest fractional value among all the fractional  $x_r$  values in the current LP solution, round this value to 1 or 0 respectively, and re-solve the LP. If rounding the variable to 1 violates the fleet availability constraint, we set it to zero. The procedure stops when all the service selection variables have integer values. The pseudo-code in Figure 7.8 summarizes this procedure.

The threshold  $\alpha$  for rounding down or up can be adjusted, but we observed during our computational tests that only services with a relatively high fractional value (e.g., around 0.7) are included in the optimal solution. We also observed that some services have a high initial fractional value but are not necessarily included in final solution. By eliminating unattractive services first, we retain the flexibility of using available ships for later choices instead of committing them early for chosen services. Therefore, our implementation first rounds down low  $x$ -values (thereby discarding some services) before rounding up  $x$ -variables with high fractional values.

For the considered problem instances, the LP-based heuristic yielded solutions that are within 5 % of optimality. In contrast, the initial upper bounds generated during the branch-and-bound process by solvers such as Gurobi and CPLEX are often quite poor for the LSP problem (e.g., the initial upper bound may select no service, thus incurring high costs for not meeting demand). Since our heuristic procedure identifies good solutions, we can use it to generate the initial upper bound to warm start exact solution methods such as branch-and-

bound. The heuristic solution can also serve as an interim recommendation to the planner if solving the problem optimally takes a long time.

In the next section, we empirically evaluate the performance of the model for practical problem instances, and also assess the benefits of our enhancements such as the valid inequalities and the LP-based heuristic procedure.

## 7.4 Computational Results

We implemented the model in C++, using the Boost Graph Library to handle the graph construction and preprocessing, and solving the LP relaxation and mixed-integer programs using Gurobi 6.0. The tests were performed on a computer with an Intel Xeon CPU X5550 2.67GHz and 24 GB RAM. When solving the mixed-integer program, we terminated the procedure when either the CPU time exceeds 12 hours (43,200 seconds) or the final gap between the final upper and lower bounds is 1 % or less, whichever occurs earlier.

### 7.4.1 Test Problems

We tested several problem instances based on the data for four common container shipping sectors - Baltic Sea, West Africa, Mediterranean, and Pacific - provided in the *Liner-lib* benchmark suite ([www.linerlib.org](http://www.linerlib.org), Brouer et al. (2014a)). For each of these sectors, the candidate service pools are generated using the mat-heuristic described in Brouer et al. (2015). We generate several service scenarios for each sector, e.g.  $Pac(1)$  and  $Pac(2)$  for the Pacific sector, where each scenario corresponds to a heuristically generated and optimized network. We also consider a combined problem instance with a larger pool of services obtained by including all the services in the individual scenarios, e.g., the instance  $Pac(1,2)$  includes the candidate services from  $Pac(1)$  and  $Pac(2)$ . (For smaller problems such as those for the Baltic sector, some services are included in multiple individual scenarios, and so the number of services in the combined instance is less than the total number of services in the individual scenarios.) The algorithm used for generating the candidate services permits constructing complex routes that visit the same port multiple times; our test problems contained several such routes. For each service, the vessel type  $v$  assigned to this service takes into account the draft limitations (i.e. larger vessels cannot visit small ports) and canal limitations (i.e., larger vessels will have to use an alternative and usually longer path) of the ports on the route. The de-

Problem Sector	No. of ports	No. of commodities	No. of vessels (classes)
<b>Baltic Sea (Baltic)</b>	12	22	6 (2)
<b>West Africa (WAF)</b>	19	38	42 (2)
<b>Mediterranean (Med)</b>	39	369	28 (3)
<b>Pacific (Pac)</b>	45	722	100 (4)

**Table 7.4:** Characteristics of test problems.

mands (commodities) have associated origin and destination ports, forecasted container traffic, and freight rate. Information for each port on the route includes name, longitude/latitude, country, geographical region, cabotage rules, unit load/unload cost, unit transshipment cost, a fixed port call cost, and a variable port call cost that can vary with vessel capacity. The fleet consists of different vessel classes with varying TEU capacities. The number of vessels varies by problem instance, with smaller instances having fewer available vessels than larger instances. For each vessel, the data includes its capacity, bunker consumption for sailing as well as idle, fees associated with traversing the Suez and Panama canals, and the time-charter rate. We calculate the costs (including the cost for each service and unit penalty for not meeting demand for each commodity) using the approach described in Brouer et al. (2014a). All demands can be transshipped at most twice. Tables 7.4 and 7.5 summarize the sizes, along various dimensions, of our problem instances.

### 7.4.2 Model Dimensions and Problem Reduction

As described in Section 7.3, our modeling and methodological framework applies different types of preprocessing methods to reduce problem size.

Table 7.5 summarizes some key dimensions of the problems and the effects of problem reduction due to trade and operational policies. For the Pacific sector problem instances, we can eliminate many variables by applying the regional policy since these problems span ports in both the Asian east coast and the west coast of the Americas. Therefore, sub-paths that cross the Pacific in both directions can be eliminated, reducing the number of variables by 15 % to 20 %. The regional policy does not apply to the other shipping sectors (and hence yields no problem reduction). The number of sub-paths shown in Table 7.5 includes only those sub-paths that remain after applying the cabotage rules. The number of variables removed includes those eliminated by determining which sub-paths are eligible for each stage of transport for a commodity (see

Scenario	Sub-paths	Services	Variables after reduction	Constraints
<b>Baltic(1)</b>	55	3	515	202
<b>Baltic(2)</b>	48	3	439	167
<b>Baltic(3)</b>	30	3	162	105
<b>Baltic(1,2,3)</b>	121	7	1,329	233
<b>WAF(1)</b>	140	10	1,797	533
<b>WAF(2)</b>	160	10	2,587	701
<b>WAF(3)</b>	120	10	1,507	503
<b>WAF(1,2,3)</b>	404	24	8,727	920
<b>Med(1)</b>	283	7	15,397	5,429
<b>Med(2)</b>	324	9	19,724	6,388
<b>Med(3)</b>	406	6	30,498	7,670
<b>Med(1,2,3)</b>	1,013	22	133,220	11,990
<b>Pac(1)</b>	1,123	17	259,020	21,932
<b>Pac(2)</b>	1,224	14	315,919	27,655
<b>Pac(1,2)</b>	2,347	31	780,845	32,875

**Table 7.5:** Problem reduction and formulation sizes.

first problem reduction method in Section 7.3.5). The last two columns of Table 7.5 show the remaining total number of variables and constraints after problem reduction. All of the following computational results are based on this reduced model.

### 7.4.3 Results for the Base Model

The first four columns in Table 7.6 show the results for the base model, before adding our valid inequalities or using our heuristic solution for warm start.

We applied the cabotage rules, regional policy rules, and the first problem reduction method in Section 7.3.5 to all instances in order to ensure that we get comparable optimal solutions since these policies not only reduce problem size but also limit some undesirable flows. In general, the smaller scenarios are solved very quickly at the root node. The initial gap associated with the lower bound (defined as  $\text{Initial LB} = (\text{Final Upper Bound} - \text{Initial Lower Bound}) / \text{Final Upper Bound}$ ) is usually very small for the individual scenarios, whereas it increases for the combined scenarios that consider an extended pool of services. For *Baltic(1,2,3)*, *Med(1,2,3)* and *Pac(1,2)* the initial gap is more than 50 %. Only the *Pac(1,2)* scenario is not solved to optimality within the

Scenario	Base Model				Strengthened Model					
	Init. LB (%)	Final gap (%)	B&B nodes	CPU time (sec.)	# valid ineq.	# comp vars	Init. LB (%)	Final gap (%)	B&B nodes	CPU time (sec.)
Baltic(1)	0	0	0	0.01	22	0	0	0	0	0.01
Baltic(2)	0	0	0	0.01	19	0	0	0	0	0.01
Baltic(3)	0	0	0	0.01	16	0	0	0	0	0.01
Baltic(1,2,3)	143	0	0	0.21	245	0	53	0	0	0.13
WAF(1)	4	0	0	0.06	52	0	3	0	0	0.05
WAF(2)	3	0	0	0.13	60	12	2	0	0	0.09
WAF(3)	1	0	0	0.03	52	20	1	0	0	0.03
WAF(1,2,3)	18	0	1,722	3.93	1,288	0	9	0	1,012	3.31
Med(1)	12	0	2	0.65	1,101	178	0	0	0	0.34
Med(2)	15	0	3	0.93	1,296	126	0	0	2	1.48
Med(3)	11	0	7	3.10	1,123	37	0	0	2	1.42
Med(1,2,3)	61	0	169	543	6,939	585	7	0	46	304
Pac(1)	11	0	22	270	2,220	905	7	0	5	205
Pac(2)	19	0	3	337	6,233	1,871	1	0	0	108
Pac(1,2)	82	11	512	> 43,200	18,165	3,204	35	0	319	23,678

**Table 7.6:** Computational Results for base and strengthened model.

Init. LB % = (Final Upper Bound – Initial Lower Bound)/Final Upper Bound

Final Gap % = (Final Upper Bound – Final Lower Bound)/Final Upper Bound

# Valid inequalities = Number of valid inequalities (9) – (11) added to the model

# Comp. vars. = Number of composite variables, obtained by combining adjacent flows

time limit of 12 hours, but the gap is reduced from an initial value of 82 % to a final value at termination of 11 %. All other scenarios are solved to within 1 % of optimality within 5 minutes.

#### 7.4.4 Effect of Strengthening the Model

Table 7.6 shows that for the base model, especially for the combined scenarios, the initial lower bound is relatively weak as evidenced by the relatively high Initial LB % gap. The last six columns of Table 7.6 show the improved results using the strengthened model. The valid inequalities (7.9) - (7.11) can be added a priori before applying branch-and-bound or added as cutting planes (when violated) at intermediate stages of the solution procedure. For our test problems, we found that an effective strategy is to generate all the valid inequalities a priori but only include a selected subset in the model. Specifically, we first

add all the inequalities and solve the LP relaxation of the LSP model. Then, we retain only those inequalities that are tight (or nearly tight, with a slack not exceeding 0.1), and remove the other inequalities (that are not tight) from the model before applying the branch-and-bound procedure. As the results demonstrate, including the subset of tight valid inequalities (7.9) - (7.11) significantly improves the initial lower bound of the formulation for all instances. For most of the individual scenarios the initial gap is now at or close to 0 %, and for the larger combined scenarios the initial gap significantly improves, e.g., in *Med(1,2,3)* the initial gap decreases from 61 % to 7 %. This improvement in the initial lower bound significantly accelerates the branch-and-bound procedure; the total computational time for the strengthened model is up to 70 % lower than for the base model, and the difficult *Pac(1,2)* problem scenario can now be solved to within 1 % of optimality well within the termination time limit. Also, with the valid inequalities, fewer branch-and-bound nodes are explored, and several more scenarios are now solved at the root node. For the problem scenarios we tested, only modest reduction was possible by combining arc flow variables into composite variables since the number of nodes with in-degree or out-degree of one is limited. This reduction opportunity is likely to increase with network size, and so this approach may be promising for larger instances.

For the smallest problems, *Baltic* and *WAF*, the optimal solution of the combined scenarios uses only the services from one of the individual scenarios, but for larger problems it is beneficial to select services from different scenarios, as indicated by the results in Table 7.7. For instance, for the individual Mediterranean scenarios, the percentage of available demand that is transported is 85 %, 88 %, and 86 % in *Med(1)*, *Med(2)* and *Med(3)* respectively. By considering the combined candidate services from all three scenarios, it is possible to select a better subset of services such that the objective is improved by 17 % to 28 % compared to the individual scenarios, with the optimal solution of the combined scenario satisfying 91 % of the available demand. Similarly, for the *Pacific* problem, it is possible to improve the objective by 27 % to 45 % and increase the flow from 91 % to 98 % when we consider all the candidate services in the individual scenarios. In *Pac(1)* only 14 out of the 17 candidate services are used, and in *Pac(2)* only 12 out of 14 are used. However, in the combined *Pac(1,2)* instance, a total of 18 services are used, leading to better coverage and improved revenue. These results suggest that by combining even just a few of the scenarios generated by the heuristic to enlarge the pool of candidate services, we can obtain good overall solutions. Since the computational times are very low when there are only a few unused services, it could be worthwhile to integrate the LSP model with network design algorithms such as the one presented in Brouer et al. (2014b). This combined model could be used to periodically assess the performance and potential of the current shipping services

Scenario	No. of services selected out of candidate services	Share of available demand transported
Med(1)	7/7	84.9 %
Med(2)	8/9	88.2 %
Med(3)	6/6	86.4 %
Med(1,2,3)	7/22	90.5 %
Pac(1)	14/17	91.1 %
Pac(2)	12/14	91.1 %
Pac(1, 2)	18/31	98.1 %

**Table 7.7:** Solution characteristics of best found solutions for the larger instances using the strengthened model.

in order to eliminate or change less productive services or to assess possible addition of single strings, e.g., Plum et al. (2014), to an existing network. In both cases, the size of the possible set of services is very manageable. Also the model can test different configurations in terms of vessel classes, speed, and other characteristics of each service. In this case it is possible to include an additional constraint in the model only allowing it to select one configuration.

### 7.4.5 Heuristic Performance

For the previous problem scenarios that required more than three seconds to solve to the desired final gap, we applied our LP-based iterative rounding heuristic (described in Section 7.3.7). The first three columns of Table 7.8 show the results, namely, the quality of the heuristic solutions, measured as the heuristic upper bound relative to the final upper bound obtained using the branch-and-bound procedure, the number of rounding iterations, and CPU time for the heuristic.

As the results in Table 7.8 show, the heuristic is very quick and produces good solutions that are within 4 % of the final upper bound obtained after branch-and-bound. Only around ten rounding iterations (iterative solution of the LP relaxation) are needed, and the heuristic's computational time is a small fraction of the time needed to complete branch-and-bound.

Although solvers such as Gurobi and CPLEX can generate initial upper bounds using built-in methods, for the LSP problem, these initial upper bounds are very poor, often orders of magnitude higher than the optimal value. However, the solvers often improve this solution quickly during the branch-and-bound



Scenario	Heuristic solution gap (%)	No. of rounding iterations	Runtime heuristic (sec.)	MIP final gap (%)	B&B nodes	Runtime MIP (sec.)
<b>WAF</b> (1,2,3)	3	8	0.15	0	869	2.12
<b>Med</b> (1,2,3)	3	12	38	0	32	237
<b>Pac</b> (1)	4	3	30	0	11	200
<b>Pac</b> (2)	0	2	14	0	0	134
<b>Pac</b> (1,2)	4	11	301	0	352	22,633

**Table 7.8:** Computational results for LP-based rounding heuristic.

Heuristic gap % = (Heuristic Upper Bound – Final Lower Bound)/Final Upper Bound

MIP Final Gap % = (Final Upper Bound – Final Lower Bound)/Final Upper Bound

process. The last three columns show the effect of providing the heuristic solution value as the initial upper bound to warm start the branch-and-bound process. Compared to the results without this warm start method, the initial upper bound improves very significantly, and this approach also reduces the number of branch-and-bound nodes that need to be explored to solve the problem to within 1 % gap. The overall running time (including the time to find the heuristic solution) also decreases, although only modestly. In general, the best solution is found quickly when starting with a good solution, but proving optimality is still time consuming. For the smaller problem scenarios (not reported in the table), the heuristic procedure is able to find the optimal solution in some cases, and so all the remaining time is spent proving optimality (using the heuristic as a warm start is not effective when almost all the candidate services are included in the final solution).

Since the heuristic procedure yields solutions that are less than 5 % from optimal and does not require much computational time, this approach may be useful when planners require quick solution times to analyze different scenarios and perform sensitivity analysis.

## 7.5 Conclusions

Our Liner Service Planning model introduces a new approach to tactical network design by explicitly considering the level of service in terms of number of transshipments. The introduction of limits on transshipments for each demand gives realistic solutions in terms of cargo flow. Our modelling approach makes it possible to include many operational aspects such as cabotage rules

and regional routing policies, making the model suitable for practical use. Besides generating solutions that are implementable in practice, incorporating the trade and operational policies also reduces the model size, thereby improving computational performance. Unlike some prior models, our model can accurately account for the cost of transshipments and loading/unloading operations. The disaggregate valid inequalities that we developed significantly improve the lower bound and reduce computational time. For our largest test problem, adding the valid inequalities even permits us to solve the problem to the desired tolerance within the allotted time. Our LP-based rounding heuristic was able to find good quality solutions in relatively short time, making it suitable as a stand-alone procedure or as a means to warm start the exact solution method.

Since the computational time increases with problem size, we may need to develop other modeling approaches or further methodological enhancements to solve much larger instances. We might also consider other optimization-based approaches including path-based decomposition or heuristic strategies. These directions are worth exploring in future research.

## Acknowledgements

The authors wish to thank Professors Stefan Ropke and David Pisinger for valuable comments. Christian Vad Karsten was supported in part by The Danish Maritime Fund under the Competitive Liner Shipping Network Design project.

## Bibliography

- Agarwal, R. and Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196.
- Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11(2):186–208.
- Balakrishnan, A. and Altinkemer, K. (1992). Using a hop-constrained model to generate alternative communication network design. *ORSA Journal on Computing*, 4(2):192–205.

- Bell, M. G., Liu, X., Angeloudis, P., Fonzone, A., and Hosseinloo, S. H. (2011). A frequency-based maritime container assignment model. *Transportation Research Part B: Methodological*, 45(8):1152–1161.
- Brouer, B. D., Álvarez, J. F., Plum, C. E., Pisinger, D., and Sigurd, M. M. (2014a). A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312.
- Brouer, B. D., Desaulniers, G., Karsten, C. V., and Pisinger, D. (2015). A matheuristic for the liner shipping network design problem with transit time restrictions. In *Computational Logistics*, pages 195–208. Springer.
- Brouer, B. D., Desaulniers, G., and Pisinger, D. (2014b). A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59.
- Brouer, B. D., Dirksen, J., Pisinger, D., Plum, C. E., and Vaaben, B. (2013). The vessel schedule recovery problem (vsrp)—a mip model for handling disruptions in liner shipping. *European Journal of Operational Research*, 224(2):362–374.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Christiansen, M., Fagerholt, K., and Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation science*, 38(1):1–18.
- Gelareh, S., Nickel, S., and Pisinger, D. (2010). Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):991–1004.
- IMO (2012). International shipping facts and figures - information resources on trade, safety, security, environment. Technical report, International Maritime Organization, <http://www.imo.org/KnowledgeCentre/ShipsAndShippingFactsAndFigures/Pages/Default.aspx>.
- Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138.
- Kjeldsen, K. H. (2011). Classification of ship routing and scheduling problems in liner shipping. *INFOR: Information Systems and Operational Research*, 49(2):139–152.

- Meng, Q. and Wang, S. (2011). Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):695–708.
- Meng, Q. and Wang, S. (2012). Liner ship fleet deployment with week-dependent container shipment demand. *European Journal of Operational Research*, 222(2):241–252.
- Meng, Q., Wang, S., Andersson, H., and Thun, K. (2014). Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280.
- Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and combinatorial optimization*. Wiley-Interscience.
- Plum, C. E., Pisinger, D., Salazar-González, J.-J., and Sigurd, M. M. (2014). Single liner shipping service design. *Computers & Operations Research*, 45:1–6.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374.
- Wang, S. and Meng, Q. (2011). Schedule design and container routing in liner shipping. *Transportation Research Record: Journal of the Transportation Research Board*, (2222):25–33.
- Wang, S. and Meng, Q. (2012). Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B: Methodological*, 46(5):615–633.
- Wang, S. and Meng, Q. (2014). Liner shipping network design with deadlines. *Computers & Operations Research*, 41:140–149.
- Wang, S., Meng, Q., and Sun, Z. (2013). Container routing in liner shipping. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):1 – 7.



# Optimal Selection of Liner Containership Services with Limited Transshipments

with Anantaram Balakrishnan<sup>1</sup>

---

## Abstract

We address a tactical planning problem facing container shipping companies of selecting which services to operate from a given pool of candidate routes so as to maximize profit. We propose a novel multi-layer multi-commodity model where we incorporate level of service requirements explicitly by limiting the number of transshipments as well as maritime cabotage rules. We solve large-scale realistic problem instances to find optimal or provably near-optimal solutions and present several heuristics based on problem characteristics that are effective at quickly producing good initial solutions.

---

<sup>1</sup>Balakrishnan, A. and Karsten, C. V. (2015). Optimal Selection of Liner Containership Services with Limited Transshipments. Submitted to *European Journal of Operations Research*.

## 8.1 Introduction

Increasing global trade, combined with the cost, energy, and environmental advantages of ocean transport compared to other modes of long-haul transport have led to significant growth in the demand for container shipping services. Seaborne international trade increased by 40 % during the past decade (UNCTAD, 2011), reaching total containerized shipment volumes of around 160 million twenty-foot equivalent units (TEU) in 2014 (UNCTAD, 2014). To meet this growing demand, shipping lines are adding capacity by commissioning new and larger ships. From 2004 to 2014, among the industry leaders, total containership capacity per company grew by more than 160 % (UNCTAD, 2014). Modern cost and energy efficient container ships built in 2015 can carry almost 20,000 TEU of cargo (up from a maximum capacity of around 9,000 TEU in 2005), but can cost more than 100 million dollars. With such high investments, utilizing the ships fully and maximizing revenue are important priorities for liner shipping companies. This paper addresses a tactical planning problem to accomplish these goals. Given the available ships and candidate services, the problem entails selecting the services to offer and deciding what port-to-port demands to serve and how to route these shipments in order to maximize profits, net of costs for operating the services, loading and unloading containers, and other expenses. Importantly, consistent with current cargo routing policies, we limit the number of permitted transshipments when deciding the container routes (this limit can vary by origin-destination and class of demand). Such restrictions have not been adequately addressed in previous models. These *level of service* requirements reflect both customers' service expectations and the routing policies of shipping companies. From a customer's perspective, having fewer transshipments reduces handling time and risk of damage and also reduces the risk of missing connections and possibly transit time. For this tactical planning problem, we propose a new and effective mixed integer programming formulation that models the container flows and transshipments (and their associated costs and constraints) over a "logical" layer while capturing service selection, loading, and capacity constraints at the "physical" layer. To effectively solve this large-scale and difficult problem, we propose several modeling and algorithmic enhancements. Unlike prior methods that require implementing specialized techniques such as column generation or use heuristic techniques, our approach exploits the capabilities of contemporary integer programming solvers for solving large-scale problem instances to find optimal or provably near-optimal solutions to the problem. We apply our model to data from the benchmark suite *Liner-lib* which among others is based on data from a container shipping line to demonstrate the practical viability and effectiveness of our approach.

The research literature on ship routing and scheduling is quite extensive. Ronen (1983, 1993), Christiansen et al. (2004, 2013), and Meng et al. (2014) classify this literature and provide comprehensive reviews of papers that have appeared over the past forty years. Broadly, the planning problems vary depending on whether the ship operations correspond to liner shipping, tramp shipping, or industrial shipping. In liner shipping, which is the focus of this paper, shipping companies operate periodic and scheduled services on specified cyclic routes; the assignment of available ships to each route determines the capacity and frequency of these services. In this context, the planning problems fall into the natural hierarchy of strategic, tactical, and operational decisions (see, for instance, Agarwal and Ergun (2008), Kjeldsen (2011), and Meng et al. (2014)). Strategic issues relate to long-term decisions on which markets to serve, whether to change the fleet size and mix, and what shipping alliances to pursue. At the other extreme, operational decisions focus on daily or weekly choices on which cargo to accept, how to route the accepted shipments, and how to reschedule operations to cope with disruptions. At the intermediate level, tactical planning entails selecting the ship routes (port rotations) to serve, assigning the available ships to each route, deciding service frequency and speed, and scheduling the port visits so as to maximize profit. These decisions on network design and fleet deployment typically have a planning horizon ranging from three months to a year, and need to be reviewed when demand patterns and operating conditions change. Since our model addresses tactical decisions, we briefly review the recent literature on related problems.

Meng and Wang (2011) consider a restricted version of the tactical problem of selecting services in a network of hub and feeder ports to meet all demand at minimum cost. Each feeder port is assigned to exactly one hub port, and containers can only transfer from one service to another at hubs. Cargo between two hubs cannot be transshipped. With these assumptions and a limit of at most two transshipments the size of the solution space is significantly reduced and we can easily enumerate the limited number of the possible paths for containers between any origin and destination. The model is solved as a mixed integer program. Christiansen et al. (2013) discuss a similar approach of selecting routes or services from a given candidate set rather than incorporating route formation decisions as endogenous choices within the tactical planning model. Level of service requirements are addressed in some more restricted tactical level planning problems such as schedule design and fleet deployment. Wang and Meng (2011) study the tactical problem of schedule design and container routing over a set of predefined paths. They consider level of service indirectly by minimizing costs that increase with transit times and solve the problem heuristically using a hybrid genetic algorithm. Wang and Meng (2012) also consider cost-minimizing schedule design subject to restrictions on (uncertain)



transit time and solve a stochastic non-linear mixed integer program using a cutting-plane approach. Meng and Wang (2012) study fleet deployment and container routing with transit time levels and solve a relaxed formulation of the problem to obtain lower bounds and use these in a global optimization algorithm. Plum et al. (2014) address the problem of designing a new service to add to a network, and propose an exact branch-and-cut-and-price algorithm that is effective for generating a single service covering up to 25 ports, which is larger than most real services which usually call 10-20 ports. Time constraints are imposed on the transported cargos on the single service considered.

The strategic liner shipping network design problem is closely related to the tactical level problems, however mainly heuristic methods for the problem have been presented. Agarwal and Ergun (2008), Álvarez (2009), Mulder and Dekker (2014) and Brouer et al. (2014b, 2015) address the issues of network design, and present heuristic methods for generating or improving a network of services that scales to larger instances, but with no guarantees on solution quality. Only very recently service level requirements have been addressed in the network design process. Brouer et al. (2015) incorporate service level requirements in the network design process by including transit time restrictions on each cargo and propose a matheuristic that iteratively improves the network by insertion/removal of ports in each service. For smaller instances of the problem some work has been done on exact methods and Reinhardt and Pisinger (2012) propose an exact branch-and-cut method for the liner shipping network design problem, but their method only solves smaller problems with up to 15 ports.

Different approaches are taken to reduce the difficulty of the network design problem and e.g. Agarwal and Ergun (2008) neglect transshipment costs in the design phase, and Álvarez (2009) does not impose frequency requirements on the services. Mulder and Dekker (2014) present an improvement heuristic where ports are aggregated into clusters to reduce the problem size. Brouer et al. (2014b, 2015) construct the initial set of services based on a greedy knapsack heuristic and then improve the single services iteratively by solving a mixed integer program to insert or remove port calls.

In another stream of research, Gelareh et al. (2010) and Agarwal and Ergun (2010) consider strategic aspects of alliances in liner shipping networks.

At the operational level there is some literature related to container routing where level of service is explicitly considered. Wang et al. (2013) incorporate level of service in container routing by formulating and solving an integer program for generating a single container path for an OD-pair taking transit time and cabotage rules into account. Karsten et al. (2015) consider container routing with transit time limits on all container paths and propose an efficient

column generation algorithm to solve the corresponding multi-commodity network flow problem with transit time constraints.

The tactical planning problem that we address is motivated by practical issues faced by planners in long-haul liner containership companies that operate services over a general network in which transshipment can occur at many ports. We refer to the above service planning problem as the *liner containership service selection problem with limited transshipments* (LSSLT problem). We propose a novel mixed integer programming model that uses a parsimonious representation of the container movements and transfers as multi-commodity flows over a logical network that permits capturing the transshipment costs and restrictions. We then assign these flows to the resources, i.e., chosen liner shipping services, in the physical network, taking into account ship capacities and operating costs. This modeling approach reduces the number of variables and constraints relative to alternative models with disaggregate commodities and flow assignment decisions. Our model can also readily accommodate a variety of operational restrictions such as cabotage rules and routing policies. To accelerate exact solution methods for the LSSLT problem, we develop problem reduction methods (to eliminate some variables and constraints a priori), propose valid inequalities to strengthen the linear programming relaxation, and apply an optimization-based and two problem specific heuristics to generate good initial solutions. Computational results using the model and solution methods for actual problem instances in the *Liner-lib* benchmark suite, which includes data provided by Maersk Line (one of the world's largest containership operators), demonstrate that our approach is effective in finding optimal or near-optimal solutions for practical problems.

The rest of this paper is organized as follows. Section 8.2 reviews the context and defines the LSSLT problem, introduces terminology and notation and presents our two-layer multi-commodity model. Section 8.3 outlines model extensions, variants, and strengthenings. Section 8.4 presents computational results and a discussion of these. Section 8.5 concludes the paper.

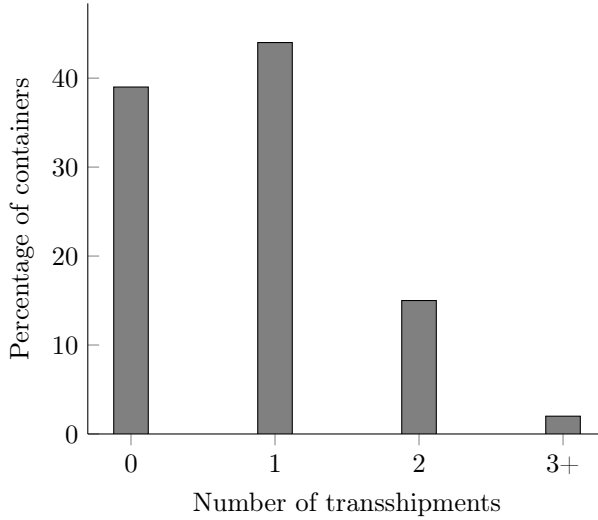
## 8.2 Problem Definition and Model Formulation

Liner shipping companies provide scheduled shipping services for container transport by dispatching vessels on various routes at periodic intervals. Each *route* is a cyclic sequence of ports on a vessel's itinerary; the vessel may visit the same port multiple times in each cycle. The frequency of service on a route, i.e., the number of times per week that assigned vessels complete the route, de-

depends on the length of the route, vessel speed, and number of vessels deployed on the route. We refer to a route with an assigned set of vessels and frequency of port visits as a *service*. A *leg* (or service leg) is the portion of a service between two consecutive ports on the route. The *capacity* of each service leg, i.e., the maximum number of containers that can be transported on that leg, depends on the frequency and types of vessels assigned to that service.

### 8.2.1 Problem Setting

The liner shipping company's demand consists of containers that need to be transported between various origin-destination port pairs. For each container that the company agrees to transport, the company must determine an itinerary or "trip plan" specifying the sequence of services on which the container is transported and the intermediate transshipment ports at which the container is transferred from one service to another. Since these inter-service *transshipment* operations require resources at the intermediate port to unload, store, and re-load the container, they are both expensive and time consuming. Consequently, trip plans that entail fewer intermediate transshipments are preferred. Moreover, customers may specify the maximum number of permitted transshipments for their containers in order to limit handling, damage, delays, and layover times, especially for hazardous and high-value cargo. On the other hand transshipment operations are important for liner shipping companies since they permit using vessel capacities effectively and having fewer transshipments reduces non-value added steps, handling capacity requirements, and transshipment costs. The practice of limiting the number of transshipments is evident from the distribution of the number of transshipments for Maersk Line, shown in Figure 8.1. The number of transshipments can vary based on the origin-destination pair. In general, intra-region cargo has fewer transshipments than inter-region shipments as might be expected, but the structure also varies across different regions as shown in Table 8.1 (observe that the number of transshipments is not symmetric, i.e., it can vary depending on the direction of shipment between two regions). In practice, although some containers can have up to five or six transshipments, most are transshipped no more than two times. These characteristics imply that the planning model must have the ability to incorporate demand-specific limits on the number of transshipments. Our model permits such limits; moreover, between the same origin-destination pair, we can distinguish between demand classes based on their maximum permitted number of transshipments. For tactical planning, we are given the forecasted weekly demand (maximum available or offered load, in terms of containers) for each origin-destination pair and demand class, and the associated revenue per container. The shipping company must decide how much of each demand to ac-



**Figure 8.1:** Histogram of number of transshipments for all services and cargoes during one week in Maersk Line’s service network configuration. The x-axis is the number of transshipments and the y-axis indicates the percentage of containers.

cept and route on its chosen service network while they meet the transshipment constraints and vessel capacities.

The focus of the tactical service planning problem is to decide what services to offer using the available fleet so as to maximize profits from its accepted cargo between various origins and destinations. Rather than designing the network (routes, vessel assignments, and service frequency) from scratch (as in the models proposed by Agarwal and Ergun (2008), Álvarez (2009), and Brouer et al. (2014b, 2015)), we model the service planning problem as one of selecting services from a candidate set of services – possible routes with associated frequency and vessel assignments – provided by the planner or an algorithm. This approach is not only quite appropriate for tactical planning but also makes the model more versatile, e.g., to capture transshipment costs and constraints. As we noted earlier, the tactical planning problem arises whenever demand patterns or operating costs and conditions change, and may be applied every six months or so. In these situations, the planner often prefers to limit the service choices and changes vis-à-vis the current service network. Moreover, the option to specify candidate services permits planners to incorporate not only economic factors but also strategic, operational, political, competitive, and

Destination	Origin					
	Africa	Asia	Australia	Europe	North America	South America
Africa		1.48	1.98	1.06	0.98	1.51
Asia	1.23		1.18	0.96	0.95	1.43
Australia	1.67	1.15		1.75	1.29	1.45
Europe	1.18	1.07	1.52		0.70	1.06
North America	1.30	1.31	0.93	0.84		1.04
South America	1.63	1.43	1.25	1.09	1.18	

**Table 8.1:** Average number of transshipments for inter-regional transport for all services and cargoes during one week in Maersk Line’s service network configuration.

other issues that may not be easy to represent in optimization models.

Given the candidate services, available vessels, and projected container transportation demand between various origins and destinations with associated transshipment limits, the LSSLT problem entails deciding which services to offer using the available vessels, how much of each demand to meet, and how to route these demands on the chosen service network so as to maximize net profit (= revenue minus operational costs for vessel usage, transshipment, etc.) while satisfying the service capacities and container transshipment limits.

### 8.2.2 Modeling Approach

A liner shipping network is generally composed of a set of services visiting various ports (nodes in a graph) that are connected by sailing legs of the service (arcs in a graph). Goods in containers can be transported between any pairs of ports and if the origin and destination port is not serviced by the same vessel it can be transshipped between services at intermediate ports where they meet. The capacity of each arc is determined by the size of the vessel deployed. A straightforward modeling of the LSSLT problem is to use a multi-commodity formulation that uses the sailing legs as arcs and represents container flows on the sailing legs and use binary variables to indicate which services are used. However, the number of transshipments in such a formulation is hard to capture as it requires adding integer variables and forcing constraints to indicate when the container is transferred from one service to another. An alternative approach is to use an augmented network based on sub-paths as arcs as done in Balakrishnan and Karsten (2015). A sub-path is a portion of a ship’s service

between any two ports of call (not necessarily consecutive) on the corresponding route. Hence, if a service visits  $n$  ports, there can be as many as  $n(n - 1)$  associated sub-paths on that service leading to a large number of potential sub-paths for a large pool of services.

To model the LSSLT problem, we propose a new multi-commodity and multi-level mixed-integer formulation with binary variables for service selection and continuous variables for container routing and service loading/assignment. Among its innovative features are a judicious choice of commodities to reduce the number of variables, the separation of container routing and service assignment decisions on two separate network layers (logical versus physical), and the representation of transshipment limits via suitable definition of the (logical) network and appropriate container flow decision variables to strengthen the model. We next provide an intuitive discussion of these features and our modeling strategy before presenting the formal mathematical model.

A route is a cyclic sequence of ports visited by a vessel. The same port may be visited more than once by the vessel (i.e., cycles need not to be simple). A service is a route assigned a number of vessels of a particular vessel type and frequency. Each service has an associated fixed cost for operating it. A portion of a container's itinerary from the port at which it is loaded on a vessel to the port at which it is unloaded from the vessel is termed a *segment*. Each segment is characterized by a starting port and an ending port; these ports can be the container's origin, destination, or intermediate (transshipment) points. A segment is defined between two ports only if there is at least one service that visits both ports. A segment is a link in the "logical" network layer; for each segment, the physical layer may contain several services that can carry the traffic on this segment. A *service segment* is a portion of a service from the starting port of a segment to the ending port of that segment. Each service segment consists of consecutive service legs from the segment's starting port to ending port. Thus, each service segment has an associated segment and service. For a given segment, the service segments corresponding to different services may pass through different intermediate ports (at which the container remains on the vessel). A *demand* is a container transportation opportunity from an origin port to a destination port with a particular level of service, i.e., specification of the maximum number of transshipments permitted for the containers of this demand. Thus, we permit multiple demands, each with different service requirements, between any origin-destination pair of ports. Each demand is characterized by origin port, destination port, maximum number of permitted transshipments, and available or offered load (number of containers).

In the following we describe the network representation in further detail. We

rely on a multi-layer representation where the container routing is handled in the logical layer of the network such that the number of transshipment can be correctly accounted for. In the physical layer of the network the cargo flow is assigned to actual sailing legs such that we ensure we have the needed capacity based on the selected services.

### 8.2.2.1 Multi-layer Representation

The physical layer of the network contains arcs corresponding to sailing legs,  $E$ , and nodes corresponding to ports,  $P$ . The logical network layer is also defined over nodes corresponding to ports; however, its arcs correspond to segments,  $A$ , for which there is at least one service that visits the two ports at the ends of this arc,  $A = \{(i, j) : i, j \in P \text{ and } i, j \in P(s) \text{ for at least one service } s \in S\}$  where  $P(s)$  is the set of ports visited by service  $s$ .

### 8.2.2.2 Container Routing vs. Service Assignment

To decide the route for each commodity, we model the commodity flows on the “logical” network layer whose arcs represent segments (pairs of ports between which a container uses a single service). To model the service selection and capacity constraints, we allocate the planned *total* flow on each segment to various chosen services that can transport the loads on the segment subject to service capacity constraints.

### 8.2.2.3 Aggregate Commodities

Instead of defining a separate commodity for each demand (O-D pair  $\langle k, l \rangle$  and service level),  $q \in Q$ , we define one commodity,  $k \in K$ , for all the containers that originate at a port node  $k$ . We then “decompose” this commodity into flows for individual O-D demands by tracking the commodity’s “outflow” from the system at various destinations. Flow conservation constraints trace paths from the origin port  $k$  of commodity  $k$  to various destination ports of demands originating at port  $k$ .

### 8.2.2.4 Max-transshipment Modeling

To model the maximum number of permitted transshipments for each demand,  $h_q$ , we use variables that are indexed by “segment sequence”, and define appropriate flow conservation equations to ensure consistency in segment sequencing.

## 8.2.3 Mathematical Model

Before presenting our model for the LSSLT problem, we introduce some additional necessary notation. For each candidate service,  $s \in S$ , there is an associated vessel type,  $t \in T$ , frequency, and a number of vessels used  $m_{ts}$  which gives the weekly capacity,  $u_s$ , of the service. Associated with each service there is a binary decision variable,  $z_s$ , indicating whether the service is selected and a corresponding cost,  $f_s$ , for selecting the service. The number of containers from each demand that are transported is tracked by the continuous variable  $v_q$  and the revenue for transporting commodity  $q$  is  $r_q$ . Table 8.2 - 8.4 summarizes the notation used for the model.

Variables	Definition
$z_s$	1 if service $s$ is selected, 0 otherwise; for all $s \in S$
$x_{ij}^{kh}$	Number of containers of commodity $k$ (from origin port $k$ ) that are transported using segment $(i, j)$ as the $h^{th}$ segment, for all $k \in K$ , $(i, j) \in A^{kh}$ , $h = 1, 2, \dots, H^k$
$y_{ij,s}$	Number of containers from segment $(i, j)$ that are transported from port $i$ to port $j$ using service $s$ , for all $(i, j) \in A$ , $s \in S_{ij}$
$v_q$	Number of containers of demand $q$ that the solution satisfies, for all $q \in Q$

**Table 8.2:** Overview of decision variables.



Params/ Sets	Definition
$P$	Set of all ports in the shipping network; $i, j, \dots, \in P$
$S$	Set of all candidate services, $s \in S$
$P(s)$	Set of all ports visited by service $s \in S$
$E(s)$	Set of all legs in service $s \in S$ ; $e \in E(s)$
$T$	Set of vessel types, $t \in T$
$M_t$	Number of vessels of type $t$ available for deployment
$m_{ts}$	Number of vessels of type $t$ required for service $s$
$u_s$	Capacity of service $s$
$Q$	Set of all demands; $q \in Q$
$o_q, d_q$	Origin and destination for demand $q \in Q$
$b_q$	Offered or available load (in containers per week) for demand $q \in Q$
$h_q$	Maximum number of transshipments permitted for containers from demand $q \in Q$
$r_q$	Revenue per container for transporting each container of demand $q$
$Q^k$	Set of all demands in $Q$ that have port $k$ as their origin; $Q^k = \{q \in Q : o_q = k\}$
$L^k$	Set of destination ports for demands originating at port $k$ ; $L^k = \{l \in P \setminus \{k\} : l = d_q \text{ for some } q \in Q^k\}$
$Q_{kl}$	Set of all demands in $Q$ that have port $k$ as origin and port $l$ as destination; $Q_{kl} = \{q \in Q : o_q = k, d_q = l\}, l \in L^k$
$H^k$	Maximum number of transshipments permitted for any demand originating at port $k$ ; $H^k = \max\{h_q : q \in Q(k)\}$
$k$	Index of commodity $k \in K$ , i.e., flows originating at port $k$
$f_s$	Fixed cost of operating service $s$
$CLO_k /$ $CUD_l$	Cost (C) for loading (L)/unloading (U) each container at origin (O) port $k$ / destination (D) port $l$
$CLT_i /$ $CUT_i$	Cost (C) of loading (L)/unloading (U) each container at transshipment (T) port $i$

**Table 8.3:** Overview of parameters and sets.

Sets	Definition
$A$	Set of all segments (ordered pair of ports) in the shipping network; $A = \{(i, j) : i, j \in P \text{ and } i, j \in P(s) \text{ for at least one service } s \in S\}$
$S_{ij}$	Set of all services $s$ that can transport traffic on segment $(i, j)$ ; $S_{ij} = \{s \in S : i, j \in P(s)\}$
$S(p)$	Set of all services visiting port $p \in P$
$E_{ij,s}$	Set of all legs in service $s \in S_{ij}$ corresponding to segment $(i, j)$
$A_{es}$	Set of all segments $(i, j)$ that can be served by service $s \in S_{ij}$ and for which the corresponding service segment contains leg $e$ ; $A_e(s) = \{(i, j) : e \in E_{ij,s}\}$
$A^{kh}$	Set of all segments $(i, j)$ that can be the $h^{th}$ segment for at least one demand $q \in Q(k)$ originating at port $k$ , for $h = 1, 2, \dots, H^k$ . For $h = 1$ (first segment), all the segments in $A^{k1}$ have node $k$ as the starting node. Conversely, none of these segments belong to $A^{kh}$ for any $h > 1$
$P^k$	Set of all ports $j \in P \setminus \{k\}$ that commodity $k$ can enter, i.e., all ports $j$ such that arc $(i, j) \in A^{kh}$ for some $h$

Table 8.4: Special sets related to the graph.

Initially we introduce a model for the LSSLT in which all demands originating at a port have the same transshipment limit, and then later extend it to multiple transshipment limits. The mixed-integer formulation of the LSSLT is:

### Objective function

$$\begin{aligned}
 \max \sum_{q \in Q} r_q v_q - & \left( \sum_{s \in S} f_s z_s + \sum_{k \in K} \sum_{(k,j) \in A^{k1}} CLO_k x_{kj}^{k1} + \right. \\
 & \sum_{k \in K} \sum_{h=1}^{H^k} \left( \sum_{(i,j) \in A^{kh}} CUD_j x_{ij}^{kh} - \right. \\
 & \left. \left. \sum_{(j,l) \in A^{k(h+1)}} (CUD_j - CUT_j - CLT_j) x_{jl}^{k(h+1)} \right) \right) \quad (8.1)
 \end{aligned}$$

**Constraints**

Commodity flow conservation on logical network:

$$\sum_{i:(i,j) \in A^{kh}} x_{ij}^{kh} - \sum_{l:(j,l) \in A^{k(h+1)}} x_{jl}^{k(h+1)} \geq 0 \quad k \in K, j \in P^k, h = 1, 2, \dots, H^k \quad (8.2)$$

Demand selection:

$$\sum_{h=1}^{H^k} \left( \sum_{i:(i,j) \in A^{kh}} x_{ij}^{kh} - \sum_{l:(j,l) \in A^{k(h+1)}} x_{jl}^{k(h+1)} \right) = v_q \quad k \in K, j \in P^k, \\ q : o_q = k, d_q = j \quad (8.3)$$

Available load:

$$v_q \leq b_q \quad q \in Q \quad (8.4)$$

Segment service allocation:

$$\sum_{k \in K} \sum_{h=1}^{H^k} x_{ij}^{kh} = \sum_{s \in S_{ij}} y_{ij,s} \quad (i, j) \in A \quad (8.5)$$

Service selection forcing and service leg capacity:

$$\sum_{(i,j) \in A_{es}} y_{ij,s} \leq u_s z_s \quad s \in S, e \in E(s) \quad (8.6)$$

Vessel availability:

$$\sum_{s \in S} m_{ts} z_s \leq M_t \quad t \in T \quad (8.7)$$

Non-negativity and integrality:

$$x_{ij}^{kh}, y_{ij,s} \geq 0 \quad k \in K, h = 1, 2, \dots, H^k, (i, j) \in A^{kh}, s \in S \quad (8.8)$$

$$z_s \in \{0, 1\} \quad s \in S \quad (8.9)$$

The objective function (8.1) captures the net revenue, defined as the total revenue for the selected demand (first term) less the costs for assigning each container to service legs from its origin to destination, loading containers at the origin, unloading and loading at each intermediate transshipment port, and unloading at the destination. Often a service penalty is incurred if a demand is not transported (e.g. Brouer et al. (2014a)). Here, this is directly included in

the revenue  $r_q$  as we maximize profit and do not need to meet demand. Hence transporting a demand means that we avoid the penalty. The commodity flow conservation constraints (8.2) trace paths from the origin port  $k$  of commodity  $k$  to various destination ports of demands originating at port  $k$ . The demand selection constraint (8.3) determines the number of containers of each demand that the solution transports, based on the number of transshipments (service level) when the container reaches the destination port  $l$ . Constraints (8.4) impose, for each demand, the available load as the upper bound on the satisfied demand. Hence we allow the model only to select profitable demands with a feasible path such that all demands and ports need not to be serviced. Given the total flow of commodities, over all commodities, on each segment in the logical layer, the segment service allocation constraint (8.5) allocates this traffic to the various service segments of the physical layer that can carry this flow. Constraints (8.6) impose the capacity constraint for each service leg while also ensuring that a service can be used only if this service is selected. Constraints (8.7) ensure that the number of vessels of each type needed for the different chosen services does not exceed the available vessels, while constraints (8.8) and (8.9) enforce non-negativity and integrality.

In the general case where demands originating at a port have the different transshipment limits, i.e.  $|Q_{kl}| \neq 1$  for any  $k \in K$ , and  $l \in P^k$  with  $Q_{kl} \neq \emptyset$  and we can introduce new  $w$ -variables where  $w_l^{kh}$  is the number of containers of commodity  $k$  that reach their destination port  $l$  using  $h$  segments, for all  $l \in L^k$ ,  $h = 1, 2, \dots, H^k$ . Then we can replace the objective (8.1) as well as constraints (8.2) and (8.3) with the following:

### Objective function

$$\begin{aligned} \max \sum_{q \in Q} r_q v_q - & \left( \sum_{s \in S} f_s z_s + \sum_{k \in K} \sum_{h=1}^{H^k} \sum_{l \in L^k} (CLO_k + CUD_l) w_l^{kh} + \right. \\ & \sum_{k \in K} \sum_{j: (k,j) \in A^{k1}} CUT_j (x_{kj}^{k1} - w_j^{k1}) + \\ & \left. \sum_{k \in K} \sum_{h=2}^{H^k} \sum_{(i,j) \in A^{kh}} (CLT_i x_{ij}^{kh} + CUT_j (x_{ij}^{kh} - w_j^{kh})) \right) \quad (8.10) \end{aligned}$$

### Constraints

Commodity flow conservation on logical network:

$$\sum_{i:(i,j) \in A^{kh}} x_{ij}^{kh} - \sum_{l:(j,l) \in A^{k(h+1)}} x_{jl}^{k(h+1)} = w_j^{kh} \quad k \in K, j \in P^k, h = 1, 2, \dots, H^k \quad (8.11)$$

Demand selection:

$$\sum_{q \in Q_{kl}: h_q \leq h} v_q \leq \sum_{h'=1}^h w_l^{kh'} \quad k \in K, l \in P^k \text{ with } Q_{kl} \neq \emptyset, h = 1, 2, \dots, \max_{q \in Q_{kl}} h_q \quad (8.12)$$

Non-negativity:

$$w_j^{kh} \geq 0 \quad k \in K, h = 1, 2, \dots, \max_{q \in Q_{kj}} h_q, j \in L^k \quad (8.13)$$

And constraints (8.4)-(8.9).

The new objective function (8.10) still captures the net revenue, defined as the total revenue for the selected demand (first term) less the costs for assigning each container to service legs from its origin to destination, loading containers at the origin, unloading and loading at each intermediate transshipment port, and unloading at the destination. The commodity flow conservation constraints (8.11) trace paths from the origin port  $k$  of commodity  $k$  to various destination ports of demands originating at port  $k$ . The demand selection constraint (8.12) determines the number of containers of each demand that the solution transports, based on the number of transshipments (service level) when the container reaches the destination port  $l$ . It specifies that the sum of demands from  $k$  to  $l$  satisfied for all the demand classes requiring a maximum of  $h$  transshipments must not exceed the total flow that reaches port  $l$  within  $h$  or fewer transshipments. Constraints (8.13) are the variable domains.

In the special case where all demands originating at a port have the same transshipment limit, i.e.  $|Q_{kl}| = 1$  for any  $k \in K$ , and  $l \in P^k$  with  $Q_{kl} \neq \emptyset$  the constraint (8.12) can be replaced with:

$$v_q \leq \sum_{h'=1}^{h_q} w_l^{kh'} \quad q \in Q^k, l \in P^k, k \in K.$$

Our problem formulations maximizes the same objective as used in the reference model by Brouer et al. (2014a), but should be adjusted by a constant to get the

same value as we include the penalty as an opportunity. This can be shown to be equivalent to incurring a penalty. The number of variables and constraints in the model without the  $w$  variables is smaller than in the general model, but as mentioned it only applies to the case where all demands originating at a port have the same transshipment limit.

Our model can readily incorporate origin-specific routing options, i.e., we can prohibit containers originating at a port  $k$  from being transported on specified segments (by suitably restricting the set  $A^{kh}$ ). Additionally we can model separate flows for loaded and empty containers (e.g., to capture their different routing options, capacity usage, etc.) by defining additional commodities, one for each container type. The container-to-leg assignment costs can include various costs including a cost penalty that increases with voyage time for the shipping leg. We can include a port-dependent cost per container to account for transshipment time at any port; this cost can also vary by inbound and outbound segment (but not the specific inbound and outbound service used to transport a container).

### 8.2.3.1 Assumptions

Following industry practice, we rely on a few assumptions to help reduce the model size. The cost for loading a container at its origin and for loading or unloading at transshipment points does not depend on the container's destination. The container traffic on any segment can be carried on any service corresponding to this segment, i.e., we do not permit situations in which a particular O-D demand cannot be carried on a particular service. A vessel's capacity is only limited by the number of containers it can carry (e.g., not the total weight of containers). So, we do not need to distinguish between loaded and empty containers. We can extend the model to incorporate additional dimensions of capacity (e.g., total weight carried), but possibly at the expense of adding more variables (e.g., to distinguish flows of loaded and empty containers).

## 8.2.4 Operational Policies

Both embargoes and maritime cabotage rules restrict the internal transport of goods within countries and are an important aspect of the operation and design of the network for all global liner shipping companies. Cabotage rules apply to loading and unloading of cargo within certain countries but also apply to transshipment of goods. We incorporate cabotage rules by removing

appropriate flow variables, thereby also reducing the size of the model.

Our model can also account for various routing policies. For instance, a regional policy that imposes rules such that we avoid containers taking unrealistic detours. One rule introduced in Balakrishnan and Karsten (2015) is to avoid flow that leaves the origin region if it has its origin and destination in the same region. This rule is easy to implement in our segment based formulation as we simply need to remove service segments  $(i, j)$  with:

*Origin  $i$  and destination  $j$  in the same geographical region, but include an intermediate port in another geographical region.*

Such a policy will in addition to a more realistic flow also lead to a smaller model as variables corresponding to removed segments will be eliminated.

### 8.2.5 Complexity

**Proposition:** The LSSLT problem is strongly NP hard when candidate services are overlapping and transshipments are permitted.

**Proof:** To establish this result, we transform the strongly NP-hard  $p$ -dispersion-sum problem (Erkut, 1990; Pisinger, 2006) to the LSSLT problem. In the  $p$ -dispersion-sum problem we must select  $p$  items such that the quadratic sum  $c_{ij}x_i x_j$  is maximized, where  $x_i$  denotes whether item  $p$  is selected. For an instance of the  $p$ -dispersion-sum problem we create a set of items (services) all containing two nodes (ports); a node (port)  $i$  that is only covered by item  $i$  and a common node (a hub port)  $O$  that is included in all items (services). Each item requires one vessel and there are  $p$  vessels available. Additionally, we create a demand for each pair of items  $i$  and  $j$  with profit  $c_{ij}$ . The corresponding LSSLT problem entails selecting  $p$  services (items) so as to maximize total profit from transporting cargoes; for cargoes requiring transshipments from  $i$  to  $j$  we only get the profit  $c_{ij}$  if both services  $i$  and  $j$  are selected. Since the  $p$ -dispersion-sum problem is strongly NP-hard, so is the LSSLT problem. ■

In the next section, we discuss model improvements that apply under certain circumstances and valid inequalities that strengthen the LP-relaxation of the model.

## 8.3 Model Improvements

Determining the feasible set of segments for each commodity is important to reduce the size of the model. The feasible set of segments for commodity  $k$  and transshipment sequence number  $h$  is  $A^{kh} = \cup_{q \in Q(k)} A_q^h$  where  $A_q^h$  is the set of all segments that can be the  $h^{th}$  segment for commodity  $q$ . It is important to keep the size of set  $A_q^h$  as small as possible by eliminating from this set all segments that cannot be the  $h^{th}$  segment for commodity  $q$ .  $A_q^h$  is easy to determine for  $h = 1, 2$ , and  $3$ ; for higher values of  $h$ , enumeration is still a possibility.

### 8.3.1 Strengthening the Model

In the following sections, we present several families of inequalities that are valid for the LSSLT and strengthen the LP relaxation. We present inequalities both in the logical and physical layer.

### 8.3.2 Forcing Constraints

The first class of inequalities are forcing constraints that enforce service selection and flow on segments and underlying sailing legs. They directly link the flow variables in the logical layer and the capacities in the physical layer, which are only linked indirectly through the allocation variables  $y_{ij,s}$  and constraints (8.5) in the model. The inequalities ensure that the flow originating at  $k$  must on all segments be less than the capacity of the services serving these segments. Furthermore, they can be strengthened as flow originating at  $k$  on all segments must be less than both the capacity and the demand of commodity  $k$ :

$$\sum_{h=1}^{H^k} x_{ij}^{kh} \leq \sum_{s \in S_{ij}} \min(u_s, \sum_{q \in Q^k \setminus Q'(i,j)} b_q) z_s \quad k \in K, (i,j) \in A \quad (8.14)$$

where  $Q'(i,j)$  is the set of demands that cannot flow on segment  $(i,j)$ , i.e., the eligible amount of commodity  $k$  on segments leaving a destination for one of the demands in commodity  $k$  is smaller than the total available load of commodity  $k$  and similarly a commodity is only eligible on segments that can be part of a  $h_q$ -segment path. When the available load of a commodity is smaller than the capacity, these are not implied by the original formulation. Furthermore, the



flow on arcs leaving the origin port  $k$  can only leave the origin on segments with  $h = 1$ , and only on services visiting this port. Hence, the following inequalities are also valid:

$$\sum_{i:(k,i) \in A} x_{ki}^{k1} \leq \sum_{s \in S(k)} \min(u_s, \sum_{q \in Q^k} b_q) z_s \quad k \in K \quad (8.15)$$

Additionally, we can introduce a subset of ports  $P_u \subseteq P$  that are only present in one service. For these ports, one service must necessarily carry all containers from or to this port, and we can add the following forcing inequalities based on flow originating and destined for this port on the single service.  $K(o(p_u))$  is the set of commodities originating in the port  $p_u \in P_u$  only visited by the service  $s \in S(p_u)$ , and  $K(d(p_u))$  is the set of commodities destined for the port  $p_u \in P_u$  only visited by the service  $s \in S(p_u)$ .

The flow originating in a port  $p_u \in P_u$  must satisfy:

$$\sum_{i:(k,i) \in A} x_{ki}^{k1} \leq \min(u_s, \sum_{q \in Q^k} b_q) z_s \quad k \in K(o(p_u)), s \in S(p_u), p_u \in P_u \quad (8.16)$$

The flow destined for a port  $p_u \in P_u$  must likewise satisfy:

$$\sum_{i:(i,p_u) \in A} \sum_{h=1}^{H^k} x_{ip_u}^{kh} \leq \min(u_s, \sum_{q \in Q^k} b_q) z_s \quad k \in K(d(p_u)), s \in S(p_u), p_u \in P_u \quad (8.17)$$

The inequality (8.16) is implied by (8.15), but (8.17) aggregates the flow for a particular destination and can strengthen the model. Hence, we only add (8.17).

### 8.3.3 Cover Inequalities

The fleet availability constraint (8.7) in LSSLT gives rise to a set of cover inequalities (e.g., Conforti et al. (2014)). Since all services in  $s \in S$  are feasible rotations, we can assume without loss of generality that  $m_{ts} z_s \leq M_t$  for all  $s \in S, t \in T$  and hence the knapsack set for a vessel class is:

$$V_t := \{z \in \{0, 1\}^S : \sum_{s \in S} m_{ts} z_s \leq M_t\}$$

And has dimension  $|S|$ . For each  $t \in T$ , we can define a cover inequality. A cover is a subset  $C^t \subset S$  such that:

$$\sum_{s \in C^t} m_{ts} > M_t$$

The cover is minimal if:

$$\sum_{s \in C^t \setminus \{j\}} m_{ts} \leq M_t \quad j \in C^t$$

and the cover inequality:

$$\sum_{s \in C^t} z_s \leq |C^t| - 1$$

is valid for  $\text{conv}(V_t)$ .

### Separation

Given a solution,  $\bar{z}_s \in [0, 1]$ , a violated cover inequality for a given  $t$  can be identified (if it exists) by the following integer program, IP, since  $m_{ts}$  and  $M_t$  are integers:

$$\begin{aligned} \theta &= \min \sum_{s \in S} (1 - \bar{z}_s) x_s \\ \text{s.t. } &\sum_{s \in S} m_{ts} x_s \geq M_t + 1 \\ &x_s \in \{0, 1\}, s \in S \end{aligned}$$

If  $\theta \geq 1$  the solution  $\bar{z}$  will satisfy all cover inequalities, while if  $\theta < 1$  there exist a violated cover inequality. The optimal solution to the IP, which is a knapsack problem itself, is always a minimal cover. The IP is NP-hard to solve in general, but using dynamic programming it can be solved very efficiently. Gu et al. (1998) show that it may not be attractive to find a minimal cover using the IP so we use a coefficient-independent cover generation algorithm and use the “bang for the buck” ratio,  $\text{bbr} = \frac{1 - \bar{z}_s}{m_{ts}}$  (Kellerer et al., 2004) to generate the covers:

**for all** knapsack constraints ( $v \in V$ ) **do**  
  Set  $x_s = 0$  for all  $s \in S$  where  $\bar{z}_s = 0$   
  Set  $x_s = 1$  for all  $s \in S$  where  $\bar{z}_s = 1$   
  Sort remaining according to *bbr* in increasing order  
  Set  $x_s = 1$  until  $\sum_{s \in S} m_{ts} \geq (M_t - \bar{b})$

where  $\bar{b}$  is the sum of the weights corresponding to the elements with  $x_s = 1$ . The cover  $C$  constructed this way may not be minimal, but we can delete items until the set becomes minimal after sorting the items in increasing order of  $m_{ts}$ .

### Lifting

For a minimal cover  $C$ , the cover inequality associated with  $C$  can be lifted as follows:

$$\sum_{s \in S} z_s + \sum_{s \in S \setminus C} \alpha_{ts} z_s \leq |C^t| - 1$$

We use a sequence-independent lifting procedure (Conforti et al., 2014) for the minimal cover inequalities generated.

## 8.3.4 Heuristics

To get good initial solutions we devise three heuristics. The procedures can also be used if good quality solutions are desired quickly as part of an interactive decision support tool. The first is generic and based on the LP solution whereas the two others are based on a reduction of the commodity set and tighter limits on the number of transshipments respectively.

### 8.3.4.1 LP-based Heuristic

A simple heuristic to obtain a good starting solution to the LSSLT problem, introduced in Balakrishnan and Karsten (2015), is to apply an LP based rounding heuristic that iteratively rounds the fractional service selection variables based on the LP relaxation of the problem. During each iteration, we select the lowest or highest fractional value and round this value to 0 or 1, and re-solve the LP. If rounding service selection variable to 1 violates the fleet availability based on the previously selected services it is set to zero. The pseudo-code in Figure 8.2 summarizes this procedure.

**Require:** a solution to the LP-relaxation of LSSLT  
**while**  $\mathbf{z}$  is fractional **do**  
    fix integer elements of  $\mathbf{z}$  and select smallest fractional element  $z_s$   
    **if**  $z_s < \alpha$  **then** fix  $z_s$  to 0  
    **else** pick the largest fractional element  $z_l$   
        **if** feasible in terms of fleet availability **then** fix  $z_l$  to 1  
        **else** fix  $z_l$  to 0  
    resolve LP-relaxation

**Figure 8.2:** LP-based heuristic.

The threshold for rounding down or up,  $\alpha$ , can be adjusted, but in general we observe that only services with a relatively high fractional value should be included in the set of selected services. Generally, rounding based heuristics are expected to perform better for tighter formulations.

#### 8.3.4.2 Reduced Commodity Set Based Heuristic

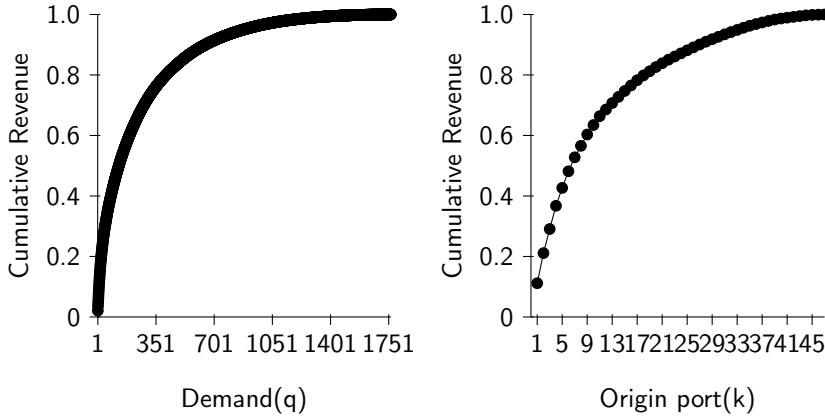
The transportation of containers between the main ports in the world constitutes a significant part of the cargo transported both in terms of volume and revenue. As seen in Figure 8.3 (left) the Pareto principle applies to the revenue distribution over demands as approximately the 20 % largest O-D pairs in terms of revenue together constitute almost 80 % of the total revenue, hence these demands will be a significant determinant in the final network design and good solutions can be expected by only considering these. However, some minor ports may be excluded as these seem less profitable. In our multi-layer model the variables are indexed by origin port,  $k$ , and a similar effect is seen by considering potential revenue as a function of origin port, Figure 8.3 (right). Hence a good quality initial solution can be expected by solving the LSSLT problem, and other liner shipping planning problems, for a reduced commodity set. In the WorldSmall instance we can potentially obtain 85 % of the revenue by only considering less than half of the commodities (origin ports). As there are demands going from this reduced set of origin ports to most destination ports, it may still be profitable to include most ports in an initial solution even though they do not appear as an origin port. The following summarizes the procedure we use:

##### **Pareto heuristic**

**Identify** the commodities (origin ports and corresponding demands) needed to (if everything is transported from these port) obtain 85 % of the potential revenue

**Solve** the LSSLT for this reduced commodity set.

We also tested a version where we include the 85 % most profitable demands instead of commodities in the model. The performance of this approach is inferior for this modeling.



**Figure 8.3:** The relative cumulative revenue for WorldSmall instance in *Linerlib* obtained as function of demands (left) and by origin port (right). Both have been sorted in ascending order. The trend is similar for the actual demand list used by one of the major carriers, Maersk Line.

#### 8.3.4.3 Transshipment-based Heuristic

Along the same lines of thought as the heuristic based on a reduced commodity set we can expect to find good solutions if we only consider a restricted version of the flow. To have a good initial solution for the model we present a heuristic where we solve a sequence of simpler problems such that flow and services are iteratively determined. In the simple case where all demands have the same number of allowed transshipments, the procedure can be described by the following pseudo code.

##### Transshipment-based heuristic

**for**  $h = 1, 2, \dots, \max\{H^k : k \in K\} - 1$

(Re-)solve the  $h$  formulation of the LSSLT model (using the  $h - 1$  solution as initial solution).

This procedure is especially beneficial when the changes in the flow solution are relatively small when allowing an additional transshipment as, most of the binary (service selection) variables will not change and most flow is already determined optimally. The number of variables increases exponentially with the number of allowed transshipments, but the solution can generally be expected to have diminishing changes such that the accuracy of method increases. On the other hand it is very efficient to obtain the solution for all direct and one-

Problem Scenario	No. of ports	No. of commodities	No. of vessels (classes)	No. of services in pool
Baltic Sea (Baltic)	12	22	6 (2)	7
West Africa (WAF)	19	38	42 (2)	24
Mediterranean (Med)	39	369	28 (3)	21
Pacific (Pac)	45	722	100 (4)	31
WorldSmall (WS)	47	1,764	263 (6)	55

**Table 8.5:** Characteristics of test problems.

transshipment flows as only few variables are considered.

## 8.4 Computational Results

The model has been implemented using the Gurobi 6.0 C++ interface. The tests were performed on an Intel Xeon CPU X5550 2.67GHz with 24 GB RAM. The termination criteria was set to a maximum CPU time of 1 hour or 1 % final gap (between the final upper and lower bounds), whichever occurs earlier. In the following all demands are allowed to transship at most twice and we present computational results for the version of the model without the  $w$ -variables. We use the cost, revenue and penalty data provided in the benchmark suite *Linerlib*, see Brouer et al. (2014a) for a description. The candidate pools of services for each instance are obtained by combining several generated networks for each scenario and are obtained from the authors of (Brouer et al., 2015). The set of services includes complex butterfly routes. The same candidate services are used in Balakrishnan and Karsten (2015), but we have extended them with a larger instance here, the WorldSmall. Table 8.5 summarizes the problem characteristics.

### 8.4.1 Model Dimensions and Problem Reduction

As described in Section 8.2, our modeling framework permits different types of preprocessing to reduce the problem size. The number of variables and constraints after model improvements are reported in Table 8.6. Furthermore the table gives a comparison with a single layer disaggregated model (Appendix 8.6), which is adapted from the model presented in Balakrishnan and Karsten (2015). The multi-layer model is much smaller both in terms of number of

Scenario	Multi-layer model			Disaggregated sub-path model		
	Segments	Variables	Constraints	Sub-paths	Variables	Constraints
<b>Baltic</b>	52	518	327	121	1,329	534
<b>WAF</b>	182	2,767	1,234	404	8,727	2,229
<b>Med</b>	624	17,106	3,882	1,174	174,952	23,056
<b>Pac</b>	920	38,651	6,581	2,347	780,845	57,134
<b>WS</b>	1,606	123,974	14,333	6,209	4,400,925	241,159

**Table 8.6:** Model size. Comparison with a single layer model with disaggregated commodities (described in Appendix 1).

Scenario	Multi-layer model				Disaggregated sub-path model			
	Initial	Final	B&B	CPU	Initial	Final	B&B	CPU
	UB gap	Gap	nodes	Time	UB gap	Gap	nodes	Time
<b>Baltic</b>	3.3 %	0.9 %	0	0.05 s	3.3 %	0.9 %	0	0.1 s
<b>WAF</b>	6.7 %	0.8 %	1,653	1.9 s	6.7 %	0.9 %	1,104	3.5 s
<b>Med</b>	21.1 %	0.0 %	337	53 s	21.1 %	0.0 %	102	775 s
<b>Pac</b>	6.3 %	0.5 %	1,092	519 s	6.3 %	10 %	0	t.l.
<b>WS</b>	6.2 %	3.3 %	1,318	t.l.	6.2 %	42 %	0	t.l.

**Table 8.7:** Computational results for the multi-layer model and the disaggregated single-layer version.

Initial UB gap % = (Initial Upper Bound –Final Lower Bound\*)/|Final Lower Bound\*|

MIP Final Gap % = (Final Upper Bound –Final Lower Bound)/|Final Lower Bound|

Final Lower Bound\* = Best lower bound obtained for the scenario

variables and number of constraints than the disaggregated model for all the considered scenarios.

## 8.4.2 Model Comparison

Table 8.7 shows the computational performance of the two basic formulations. They have the same LP-relaxation and initial upper bound gap, and a solution to the disaggregated single-layer model can be transformed directly to a solution to the aggregated multi-layer model and therefore the multi-layer model is at least as tight as the single-layer model. For the three smallest scenarios the runtime for the multi-layer model is up to 93 % faster than for the single layer model. The Pacific instance is solved well within the time limit in the multi-layer model whereas the gap is 10 % in the disaggregated single layer-model. In WorldSmall the final gap is 92 % smaller for the multi-layer model.

Scenario	Strengthened multi-layer model			
	Initial UB gap	Final gap	B&B nodes	CPU time
<b>Baltic</b>	2.8 %	0.5 %	0	0.02 s
<b>WAF</b>	4.2 %	0.9 %	416	1.2 s
<b>Med</b>	19.4 %	0.0 %	276	50 s
<b>Pac</b>	6.0 %	0.8 %	695	381 s
<b>WS</b>	5.9 %	2.5 %	1,076	t.l.

**Table 8.8:** Computational Results for the strengthened multi-layer model.

Initial UB gap % = (Initial Upper Bound - Final Lower Bound\*)/[Final Lower Bound\*]

MIP Final Gap % = (Final Upper Bound - Final Lower Bound)/[Final Lower Bound]

Final Lower Bound\* = Optimal lower bound

t.l. = time limit is set to 3,600 seconds

Table 8.8 shows results for the strengthened model where forcing constraints and covers are added. The forcing constraints are added to the model a priori, but only the inequalities that are binding when solving the LP-relaxation at the root node are kept during the branch-and-bound process. The cover inequalities are added dynamically at the root using callbacks. In Pacific the inequalities are added for services crossing the Pacific Ocean and for WorldSmall we also add inequalities for services crossing the Atlantic Ocean and the Suez Canal. The forcing constraints that are added a priori tighten the LP-relaxation of the model and the initial UB gap is reduced between 4-37 % for the considered instances. This leads to improved computational performance where between 18-74 % less nodes need to be explored and the runtimes are reduced between 6-60 % for the solved cases and the final gap is reduced by 24 % for the WorldSmall instance that is not solved to optimality within the time limit of one hour. In the WorldSmall scenarios we attain a solution with a small gap of 2.5 % and in fact the found solution is within 1 % of the value of the optimal solution. Though the disaggregated single-layer model can be significantly tightened as shown in Balakrishnan and Karsten (2015) we can still solve the multi-layer model much more efficiently.

Table 8.9 shows the solution characteristics for the optimal solution in each of the scenarios. These solutions are found by increasing the allowed time limit in the WorldSmall instance and decreasing the tolerance to  $10^{-6}$ . The results show that we get solutions where it is most profitable to transport a large part of the available demand and only a subset of the service pool is selected. The table also shows the distribution of transshipments in each of the scenarios. For the smaller scenarios only covering one region everything can be transported using up to one transshipment. The Pacific instance covers 2 main regions



Scenario	No. of services selected in optimal solution	Share of available demand transported	0 TS	1 TS	2 TS
<b>Baltic</b>	3	87.9 %	93 %	8 %	0 %
<b>WAF</b>	10	96.9 %	84 %	16 %	0 %
<b>Med</b>	5	94.2 %	55 %	39 %	6 %
<b>Pac</b>	18	98.1 %	66 %	30 %	4 %
<b>WS</b>	32	95.5 %	44 %	45 %	11 %

**Table 8.9:** Solution characteristics of optimal solutions.

TS = share of cargo transported using 0, 1, or 2 transshipments.

and still most of the containers can be transported on direct connections or with one transshipment. The WorldSmall instance shows results very much in accordance with industry practice with a distribution of the transshipments very similar compared to the data from the real Maersk Line network in Figure 8.1.

### 8.4.3 Effect of Heuristics

Table 8.10 shows the effect of the three heuristics. The LP-based rounding heuristic is relatively fast, but does not generate very good quality solutions. The pareto heuristic performs significantly better in terms of solution quality, but is relatively slow, if termination is set to be 1 % gap. The commodity sets in Baltic and WAF are so small that the heuristic is less meaningful for these instances, hence we only report results for larger instances. We included origin ports corresponding to 85 % the potential revenue. This corresponds to including 52 % of the variables and 22/47 origin ports in WorldSmall, 50 % of the variables and 15/45 origin ports in Pacific, and 70 % of the variables and 19/39 origin ports in Mediterranean. The transshipment based heuristic shows very good performance and for smaller instances it finds the optimal solution, which is not surprising given that all or most demand in these instance is transported using one or less transshipments, see Table 8.9. For the larger instances, Pacific and WorldSmall, it is able to produce very high quality solutions quickly compared to solving the full problem to optimality.

Scenario	LP-based heuristic gap	Runtime heuristic	Pareto-based heuristic gap	Runtime heuristic	TS-based heuristic gap	Runtime heuristic
<b>Baltic</b>	0.8 %	0.01 s	-	-	0.0 %	0.04 s
<b>WAF</b>	3.5 %	0.08 s	-	-	0.0 %	0.4 s
<b>Med</b>	42 %	4.8 s	3.6 %	32 s	0.0 %	17 s
<b>Pac</b>	5.4 %	16 s	4.0 %	162 s	0.6 %	119 s
<b>WS</b>	7.6 %	144 s	2.8 %	t.l.	0.6 %	t.l.

**Table 8.10:** Computational results for heuristic.

Heuristic gap = (Final Lower Bound\* - Heuristic Lower Bound)/[Final Lower Bound\*]

TS-based: Transshipment based heuristic

Final Lower Bound\* = Best lower bound

t.l. = time limit is set to 3,600 seconds

#### 8.4.4 Sensitivity to Available Demand and Fleet

Table 8.11 shows the implications of variations in the amount of available demand by  $\pm 20$  % and the number of available vessels by  $\pm 20$  %. Changes in the available amount of demand lead to two clear patterns. Revenue increases monotonically with increasing demand and the model becomes easier to solve. In the Pacific case the solution time is 772 s when only considering 80 % of the demand whereas it is only 92 s when 120 % of the demand is considered. This is not surprising as most or all cargo is transported using at most one transshipment as seen in Table 8.9. In all cases the final gap is 0.0 %. For WorldSmall the final gap achieved within one hour is 4.4 % for 80 % demand and 1.2 % for 120 % demand. The initial UB on the model increases monotonically as well with increasing demand, but the initial UB gap relative to the best found solution decreases with increasing demand.

Varying the number of available vessels shows a somewhat different picture. The initial UB and the capacity volume in TEU used does increase when increasing the number of vessels, but only until some saturation point is reached where most of the profitable demand is transported. The variation in runtimes and final gap is significantly smaller than when varying the demand. The number of vessels is divided across different vessels classes and for the considered WorldSmall instances it is clear that some reorganization of the network happens as one vessel class is more constraining than others. The number of vessels used increases with increased availability, but for WorldSmall in the vessel range 80-100 % the number of capacity miles does not increase while the number of vessels used does and more services are used. In Pacific it is also seen that when increasing to 110 % vessels become available and a service with

% dmn	# ves	IUB gap	FLB	Node count	Rt / FG	# serv used	Con. miles	Cap. miles	# ves	vol used	% trnsp.
<b>Pacific</b>											
80	100	9.1	78	1,261	772	15	1.30	1.72	73	113	97.0
90	100	7.4	89	1,395	718	14	1.49	1.88	79	116	96.7
100	100	6.0	100	695	381	18	1.63	2.09	93	141	98.1
110	100	5.5	110	548	264	16	1.76	2.17	88	134	94.5
120	100	4.6	120	61	92	16	2.26	2.82	91	143	95.5
100	81	8.7	96	841	389	13	1.58	1.94	73	111	88.7
100	91	7.8	98	574	336	15	1.52	1.87	82	119	92.6
100	100	6.0	100	695	381	18	1.63	2.09	93	141	98.1
100	109	5.9	100	667	356	18	1.63	2.08	93	139	98.4
100	119	5.9	100	792	373	18	1.63	2.08	93	139	98.4
<b>WorldSmall</b>											
80	263	8.6	79	704	4.4	28	6.55	8.04	192	395	94.9
90	263	8.3	89	997	4.6	32	7.53	9.23	215	443	94.9
100	263	6.3	100	1,070	2.5	32	8.13	9.99	217	477	94.2
110	263	5.9	110	1,099	2.6	36	8.38	10.4	239	511	94.3
120	263	4.7	120	963	1.2	36	9.05	11.2	243	540	93.1
100	209	8.6	97	727	4.7	28	8.11	10.0	199	443	90.3
100	237	6.9	99	1,215	3.0	31	8.05	10.0	218	481	93.8
100	263	6.3	100	1,070	2.5	32	8.13	9.99	217	477	94.2
100	289	6.7	99	1,224	2.9	31	8.79	10.9	222	482	94.1
100	317	6.1	101	1,128	2.7	34	8.28	10.2	234	484	94.7

**Table 8.11:** Sensitivity for the Pacific and WorldSmall scenario.

%dmn = percentage of available demand in base case

# ves = number of vessels available

IUB gap = (Initial UB - Final UB)/Final UB

FLB = Final LB divided by Final LB for base case

Rt / FG = Runtime in seconds for the Pacific scenario, Final Gap in % for WorldSmall  
(time limit is 3,600 s)

# serv used = number of services used in solution

Con. miles = Container miles (in billions) in solution

Cap. miles = Capacity miles (in billions) of network

# ves used = Vessels used in the solution

vol used = Total volume of vessels used in thousand TEU

% trnsp. = percentage of available demand transported in solution

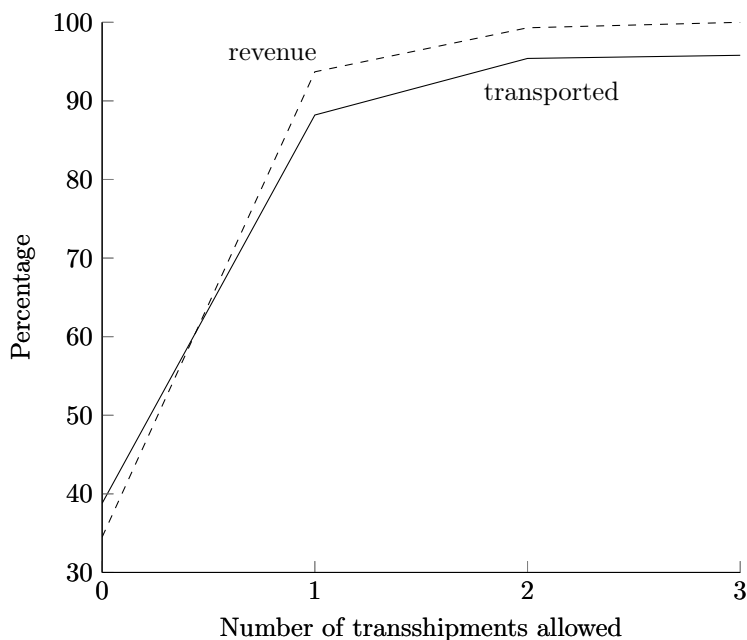
larger capacity is replaced with one with lower capacity (total TEU volume used decrease but same number of services used). The Pacific weighted capacity utilization is between 76 and 81 % and the WorldSmall weighted utilization is between 80-82 %, which is what would be expected for a realistic network due to trade imbalances. The sensitivity results also show that the algorithm generally performs well and is robust under changing conditions. The model can be used to quickly assess profit and utilization changes as a consequence of changes in the economic conditions or as a consequence of potential internal decisions.

### 8.4.5 Effect of Transshipment Limit

Finally, we look at the influence of the imposed transshipment limit. For the WorldSmall scenario Figure 8.4 shows the percentage of transported demand and corresponding relative revenue as the transshipment limit is increased. As seen, limiting the number of transshipments to two allows us to transport almost all of the demand that can be transported in this network using up to three transshipments. In the solution with three allowed transshipments only 0.7 % of the transported cargo is transported using three transshipments, while the overall amount of cargo transported is reduced by 0.1 %. The set of services changes slightly but the optimal solution still uses 32 services and the optimal objective value increases by 0.1 %.

## 8.5 Conclusions

The presented model for the LSSLT problem address a tactical planning problem facing container shipping companies of selecting which sailing routes to operate from a given pool of candidate routes so as to maximize profit. We incorporate level of service by limiting the number of transshipments, but also include cabotage rules and show it straightforward to include special regional rules. Our novel multi-layer model for the LSSLT problem performs better from a computational perspective than a previous disaggregated single-layer model for the problem. Our approach exploits the capabilities of contemporary MIP solvers for solving large-scale problem instances to find optimal or provably near-optimal solutions to the problem, and hence does not require implementing specialized techniques such as column generation or use heuristic techniques. We solve large-scale realistic problem instances to find optimal or provably near-optimal solutions in acceptable time and show that heuristics are



**Figure 8.4:** Percentage of transported demand and corresponding revenue. The revenue is normalized with the three transshipment solution.

effective at producing good quality solutions quickly. Sensitivity analyses of the fleet size and available demand shows the algorithm performs as expected and how it can be used to yield insights on profit and utilization under changing condition. Sensitivity results also show that allowing more than two transshipments only changes optimal solution insignificantly, hence in a strategic decision process it may be sufficient to consider only up to two transshipments during the network design process.

## 8.6 Appendix

We here briefly outline a single layer disaggregated multi-commodity model (Balakrishnan and Karsten, 2015) addressing the same problem as the LSSLT but with a modeling where we do not take advantage of the possibility of aggregating demands and utilizing several layers to flow demands and assign flow respectively. The model is defined over an augmented network which is similar to the logical network presented in this paper, but composed of sub-paths, where we define a sub-path as the portion of a ship's service between any two port calls (not necessarily consecutive) on the corresponding route, i.e. a sub-path is a segment with a specific route associated. Similarly to LSSLT, we define flow variables for each possible transshipment stage (a transshipment stage captures the number of previous and current transshipments that the flow has gone through). These variables are then linked across stages using appropriate flow conservation constraints. Thus, the single layer model implicitly accounts for the transshipment limits by permitting, for each demand, only as many stages as the transshipment limit for that demand as in the LSSLT. In the single layer model we do not aggregate demands based on origin port, but consider the flow of each demand separately on each sub-path. The objective and main assumptions are the same. First we introduce some additional necessary notation.

### *Variables*

$u_{ijs}^{hq}$ : number of units of demand  $q$  on sub-path  $(i, j)$  of service  $s$  as the  $h^{th}$  stage, for  $h = 1, 2, \dots, h_q$ .

### *Sets*

$A_s$ : set of sub-paths in service  $s$ ,  $a \in A^s$ .

$A_s(e)$ : set of sub-paths that use leg  $e$ .

$B_s^{hq}$ : set of sub-paths  $(i, j) \in A^s$  eligible as transshipment stage  $h$  for service  $s$  and demand  $q$ .

$P^{hq}$ : set of ports that can be reached in stage  $h$  for demand  $q$ , excluding the destination port  $t_q$ .

### *Parameters*

$c_{ijs}^{hq}$ : cost of sub-path  $(i, j)$  when used as the  $h^{th}$  stage for demand  $q$ .

Given this notation the mixed-integer model is:

$$\max \quad \sum_{s \in S} -f_s z_s + \sum_{q \in Q} \sum_{s \in S} \sum_{h=1}^{h_q} \sum_{(i,j) \in B_s^{h_q}} -c_{ijs}^{h_q} u_{ijs}^{h_q} + \sum_{q \in Q} \sum_{s \in S} \sum_{(i,j) \in B_s^{h_q}} r_q u_{ijs}^{1q} \quad (8.18)$$

Subject to:

$$\sum_{s \in S} \sum_{(i,j) \in B_s^{1q}} u_{ijs}^{1q} \leq d_q \quad q \in Q \quad (8.19)$$

$$\sum_{s \in S} \sum_{(i,j) \in B_s^{h_q}} u_{ijs}^{h_q} - \sum_{s \in S} \sum_{l: (j,l) \in B_s^{(h+1)q}} u_{jls}^{(h+1)q} = 0 \quad q \in Q, j \in P^k, h = 1, 2, \dots, h_q - 1 \quad (8.20)$$

$$\sum_{q \in Q} \sum_{h=1}^{h_q} \sum_{(i,j) \in A_s(e)} u_{ijs}^{h_q} \leq u_s z_s \quad e \in E_s, s \in S \quad (8.21)$$

$$\sum_{s \in S} m_{ts} z_s \leq M_t \quad t \in T \quad (8.22)$$

$$u_{ijs}^{h_q} \geq 0, \quad q \in Q, s \in S, (i, j) \in B_s^{h_q}, h = 1, 2, \dots, h_q \quad (8.23)$$

$$z_s \in \{0, 1\}, \quad s \in S \quad (8.24)$$

The objective (8.18) is to maximize the total profit less cost, consisting of the fixed and operating costs for the selected services, the cost of transporting goods on each sub-path and the loading/unloading or transshipment at the terminals of the sub-path. This is similar to maximizing revenue in the LSSLT as we do not need to meet demand which is secured by constraints (8.19) which consider the sub-paths incident from the origin port for a given commodity, assign flow to these sub-paths, and ensure that the flow of all demands is less than the available amount. Constraints (8.20) are the flow balance constraints for intermediate stages. It specifies that the flow of a commodity entering a node  $j$  at stage  $h$  equals the flow leaving that node for stage  $(h + 1)$ . These constraints, over all the stages, together ensure that all flows of a commodity travel from origin to destination within the specified number of stages or hops. For each commodity, we define flow variables and construct constraints (8.19) and (8.20) so as to only allow flow on paths that use less than  $h_q$  sub-paths or stages. At the same time, the model allows splitting flow on multiple origin-to-destination paths. Constraints (8.21) serve to both impose the capacity of each sailing leg and also ensures that we can assign flows to a sub-path only if the corresponding service  $s$  is selected. Constraints (8.22) specify that the total number of vessels of each type needed to operate the chosen services must not exceed the liner company's available fleet of that class.

## Bibliography

- Agarwal, R. and Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2):175–196.
- Agarwal, R. and Ergun, Ö. (2010). Network design and allocation mechanisms for carrier alliances in liner shipping. *Operations Research*, 58(6):1726–1742.
- Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11(2):186–208.
- Balakrishnan, A. and Karsten, C. V. (2015). Joint service selection and cargo routing with limited transshipments. Submitted to: Annals of Operations Research: Logistics, Optimization, and Transportation.
- Brouer, B. D., Álvarez, J. F., Plum, C. E., Pisinger, D., and Sigurd, M. M. (2014a). A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312.
- Brouer, B. D., Desaulniers, G., Karsten, C. V., and Pisinger, D. (2015). A matheuristic for the liner shipping network design problem with transit time restrictions. In *Computational Logistics*, pages 195–208. Springer.
- Brouer, B. D., Desaulniers, G., and Pisinger, D. (2014b). A matheuristic for the liner shipping network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:42–59.
- Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483.
- Christiansen, M., Fagerholt, K., and Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation science*, 38(1):1–18.
- Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer programming*, volume 271. Springer.
- Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60.
- Gelareh, S., Nickel, S., and Pisinger, D. (2010). Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):991–1004.
- Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W. (1998). Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10(4):427–437.



- Karsten, C. V., Pisinger, D., Ropke, S., and Brouer, B. D. (2015). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138.
- Kellerer, H., Pferschy, U., and Pisinger, D. (2004). *Knapsack problems*. Springer.
- Kjeldsen, K. H. (2011). Classification of ship routing and scheduling problems in liner shipping. *INFOR: Information Systems and Operational Research*, 49(2):139–152.
- Meng, Q. and Wang, S. (2011). Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):695–708.
- Meng, Q. and Wang, S. (2012). Liner ship fleet deployment with week-dependent container shipment demand. *European Journal of Operational Research*, 222(2):241–252.
- Meng, Q., Wang, S., Andersson, H., and Thun, K. (2014). Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280.
- Mulder, J. and Dekker, R. (2014). Methods for strategic liner shipping network design. *European Journal of Operational Research*, 235(2):367–377.
- Pisinger, D. (2006). Upper bounds and exact algorithms for p-dispersion problems. *Computers & operations research*, 33(5):1380–1398.
- Plum, C. E., Pisinger, D., Salazar-González, J.-J., and Sigurd, M. M. (2014). Single liner shipping service design. *Computers & Operations Research*, 45:1–6.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374.
- Ronen, D. (1983). Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, 12(2):119–126.
- Ronen, D. (1993). Ship scheduling: The last decade. *European Journal of Operational Research*, 71(3):325–333.
- UNCTAD (2011). Review of maritime transport. Technical report, United Nations Conference on Trade and Development.

- UNCTAD (2014). Review of maritime transport. Technical report, United Nations Conference on Trade and Development.
- Wang, S. and Meng, Q. (2011). Schedule design and container routing in liner shipping. *Transportation Research Record: Journal of the Transportation Research Board*, (2222):25–33.
- Wang, S. and Meng, Q. (2012). Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B: Methodological*, 46(5):615–633.
- Wang, S., Meng, Q., and Sun, Z. (2013). Container routing in liner shipping. *Transportation Research Part E: Logistics and Transportation Review*, 49(1):1 – 7.